
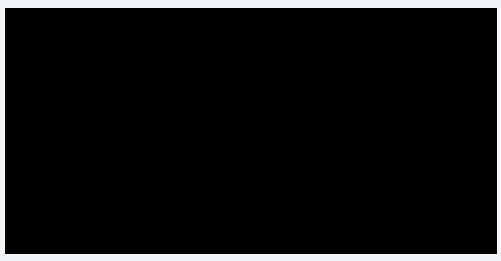


CYBER NEWS, DATA BREACH, INFOSTEALER, OSINT, VULNERABILITY

April 21, 2022

# Prynt Stealer Spotted In the Wild

## The Stealer Is New On The Cybercrime Financial Data Using A Clipper And Ke

# A New Info Stealer Pe Clipper And Keylogg

Cyble research labs discovered a new Infostealer na the cybercrime forums and comes with various capa data, this stealer can also perform financial thefts us Additionally, it can target 30+ Chromium-based brov range of VPN, FTP, Messaging, and Gaming apps. Furt functionality of this stealer.

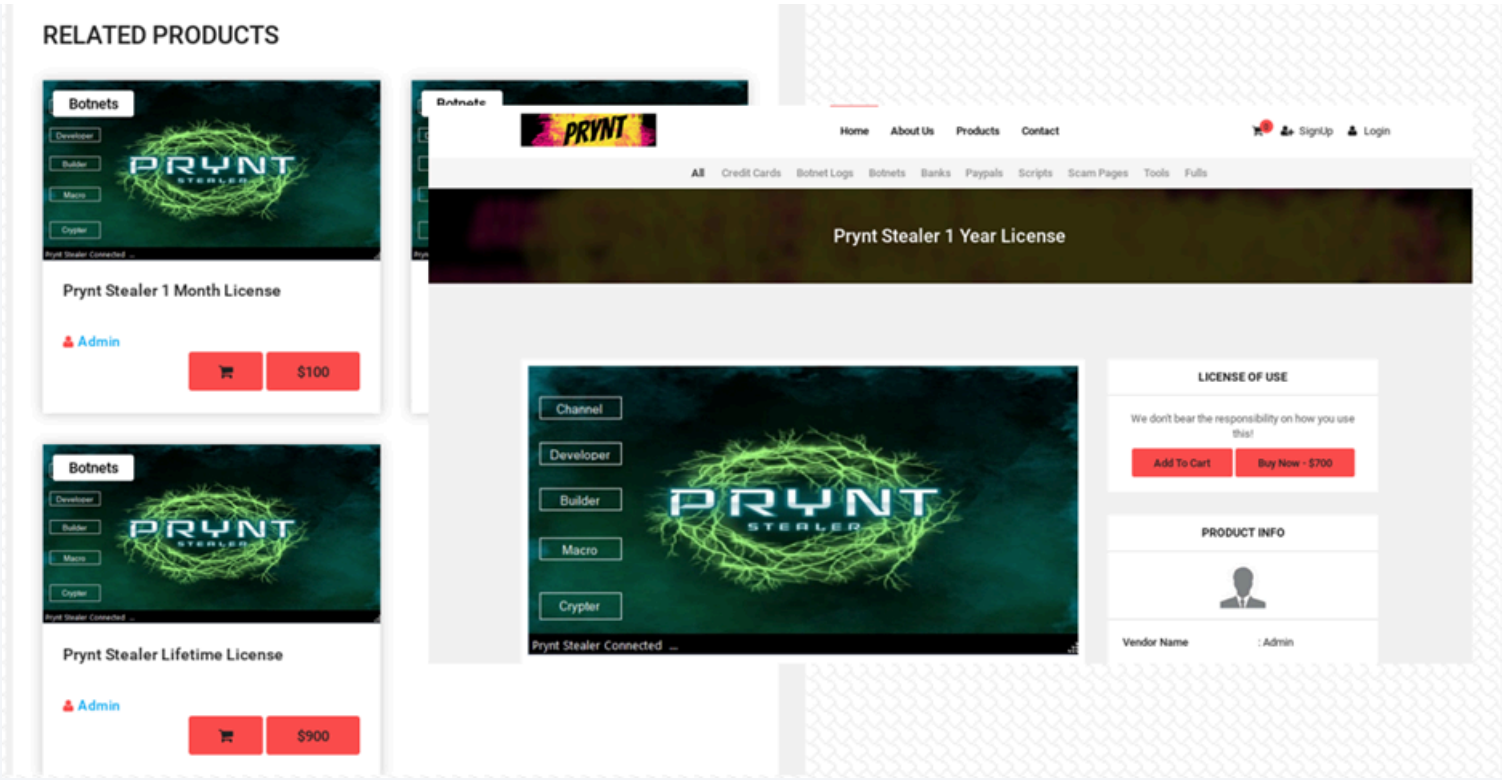


Figure 1: Post on cybercrime marketplace

The developer of the stealer recently claimed the recent versions of the stealer to be FUD (Fully Undetectable), as shown in Figure 2. We could also spot a few stealer logs available for free on the Telegram channel.

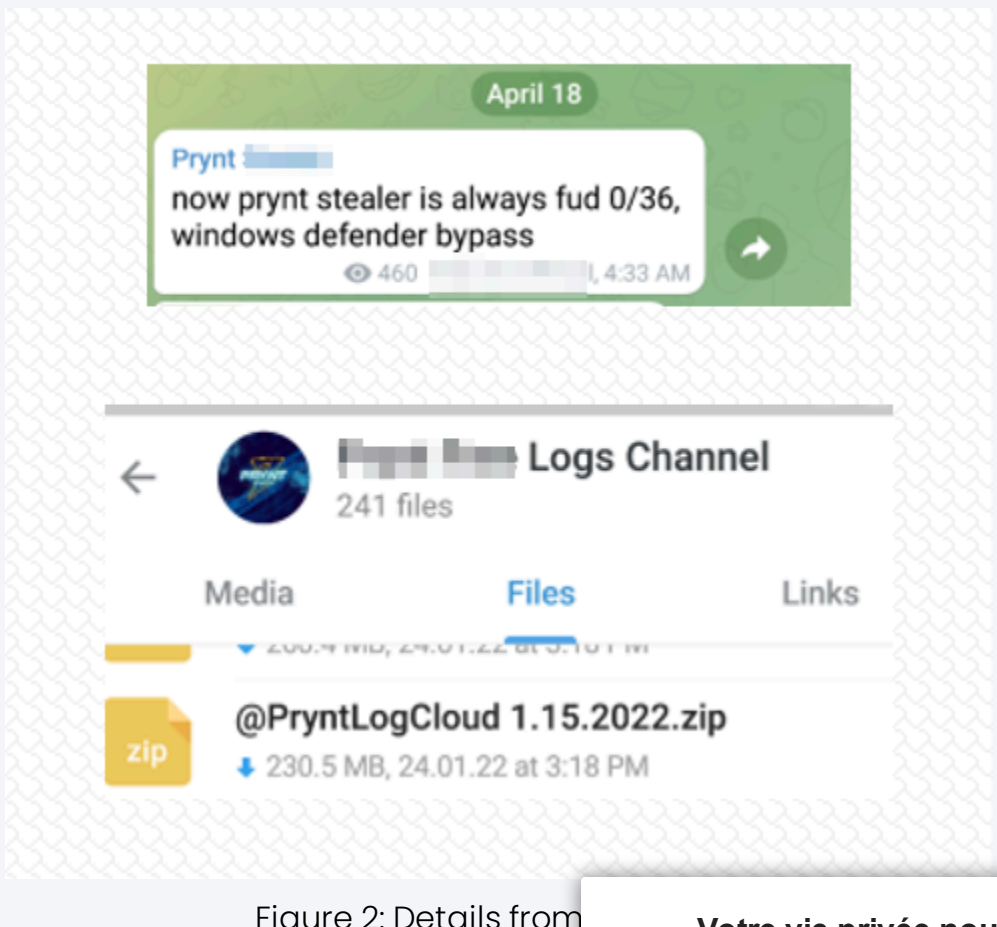


Figure 2: Details from

The embedded binary contains hardcoded strings w
Rijndael encryption algorithm. Prynt Stealer is a .Net-
details.

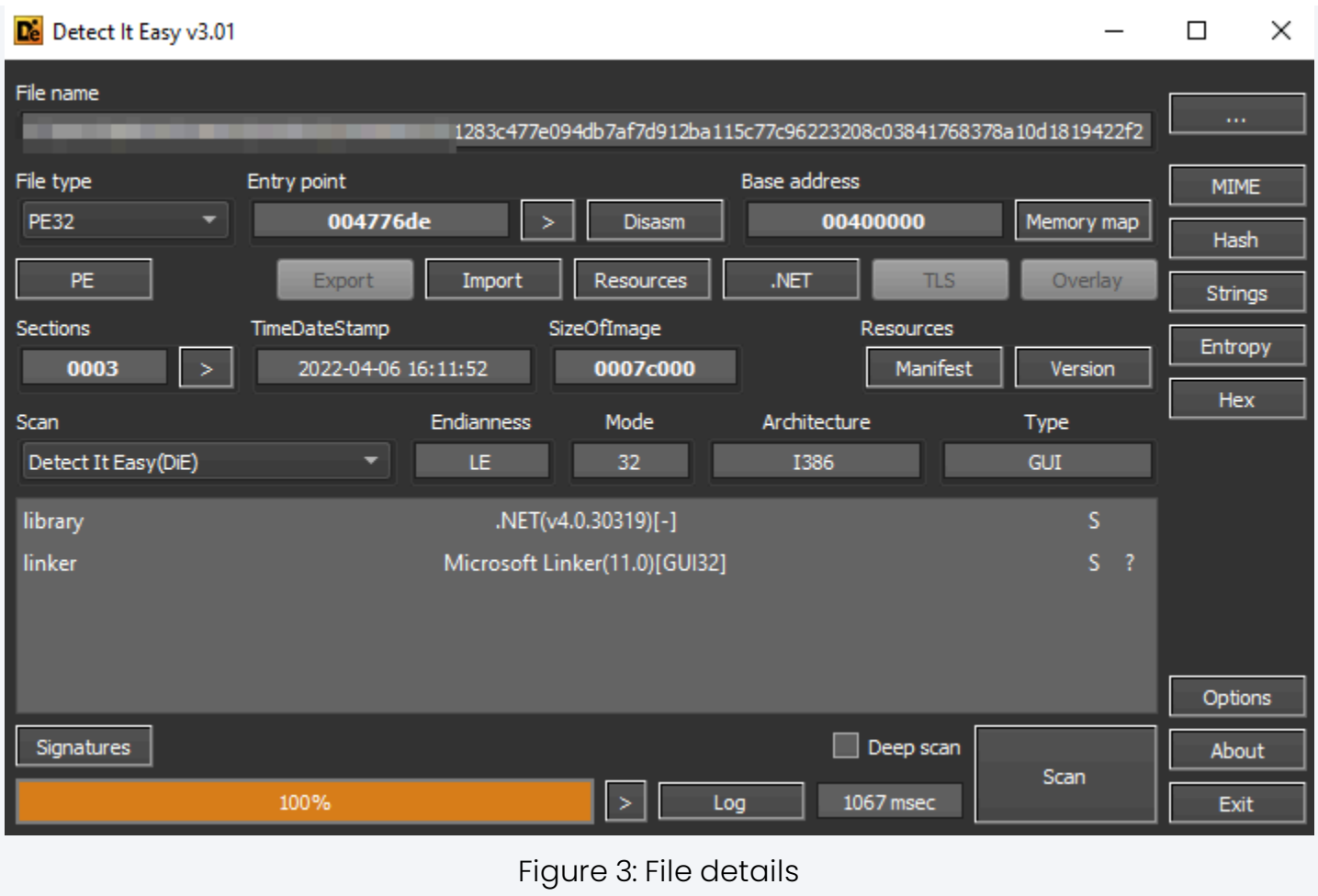


Figure 3: File details

## Technical Analysis

The sample (***SHA 256***: *1283c477e094db7af7d912ba115c77c96223208c03841768378a10d1819422f2)* has an obfuscated binary stored as a string, as shown in Figure 4.


Figure 4: Obfuscate

The binary is encoded using the rot13 cipher. ROT13 (r
one after 13 positions from the current letter. The rot13
encoded binary in this sample. The malware rather t
directly in the memory using *AppDomain.CurrentDon*

Figure 5: Binary decoding process

The malware uses *ServicePointManager* class to establish an encrypted channel to interact with the server. There are a few hardcoded strings encrypted using the AES256 algorithm. All these strings are decrypted by calling *Settings.aes256. Decrypt()* method is assigned back to the same variables, as shown in the Figure below.

Figure 6: Decrypts hardc

After this, the malware creates a hidden directory in t
using the MD5 hash value. The Figure below shows th
and hiding a directory.

Figure 7: Creates a hidden directory

Then a subfolder is created inside the parent directory created above and is named using the format "username@computername_culture." Malware will also create other folders inside this folder, such as Browsers, Grabber, etc. These folders will be used for saving the stolen data from respective sources.

The malware then identifies all the logical drives present in the victim's system using the DriveInfo() class and checks for the presence of removable devices. Next, the malware adds the drive's name and path to its target list for stealing data. After identifying the drive details, the malware steals the files from the targeted directories, as shown in Figure 8. The malware uses a multithreading approach for stealing the files fast from the victims' machines. Prynt Stealer only steals the files whose size is less than 5120 bytes and should have the following extensions:

Document: pdf, rtf, doc, docx, xls, xlsx, ppt, pptx, indd, txt, json.

Database: db, db3, db4, kdb, kdbx, sql, sqlite, mdf, mdb, dsk, dbf, wallet, ini.

Source Code: c, cs, cpp, asm, sh, py, pyw, html, css, php, go, js, rb, pl, swift, java, kt, kts, ino.

Image: jpg, jpeg, png, bmp, psd, svg, ai.

Figure 8: Steal

## Browsers

After stealing files from the victim's system, Prynt Ste

Targeted browsers include:

- Chromium-based browsers
- MS Edge
- Firefox-based browsers

**Chromium-based browsers**:

It first creates a folder named "Browsers" and then ch
the **Figure** below) in the "AppData" folder using *Direc*
malware starts stealing data from the respective loc
chromium-based browsers, as can be seen in the Fig
multiple .sqlite files for storing users' data.

Figure 9: Targeted chromium-based browsers

It steals the master key from the “Local Sate” file, which is used for decrypting the sensitive information stored in the browsers.

The malware steals Credit Cards, Passwords, Cookies, Autofill, History, Downloads, and Bookmarks data from browsers, and saves the stolen data in respective text files created under the “Browsers\Browser_Name\” directory.

Files targeted by malware for stealing data:

- Web Data (for Autofill data)
- Login Data (for Login Credentials)
- History (for search history)
- Cookies (for browser Cookies)

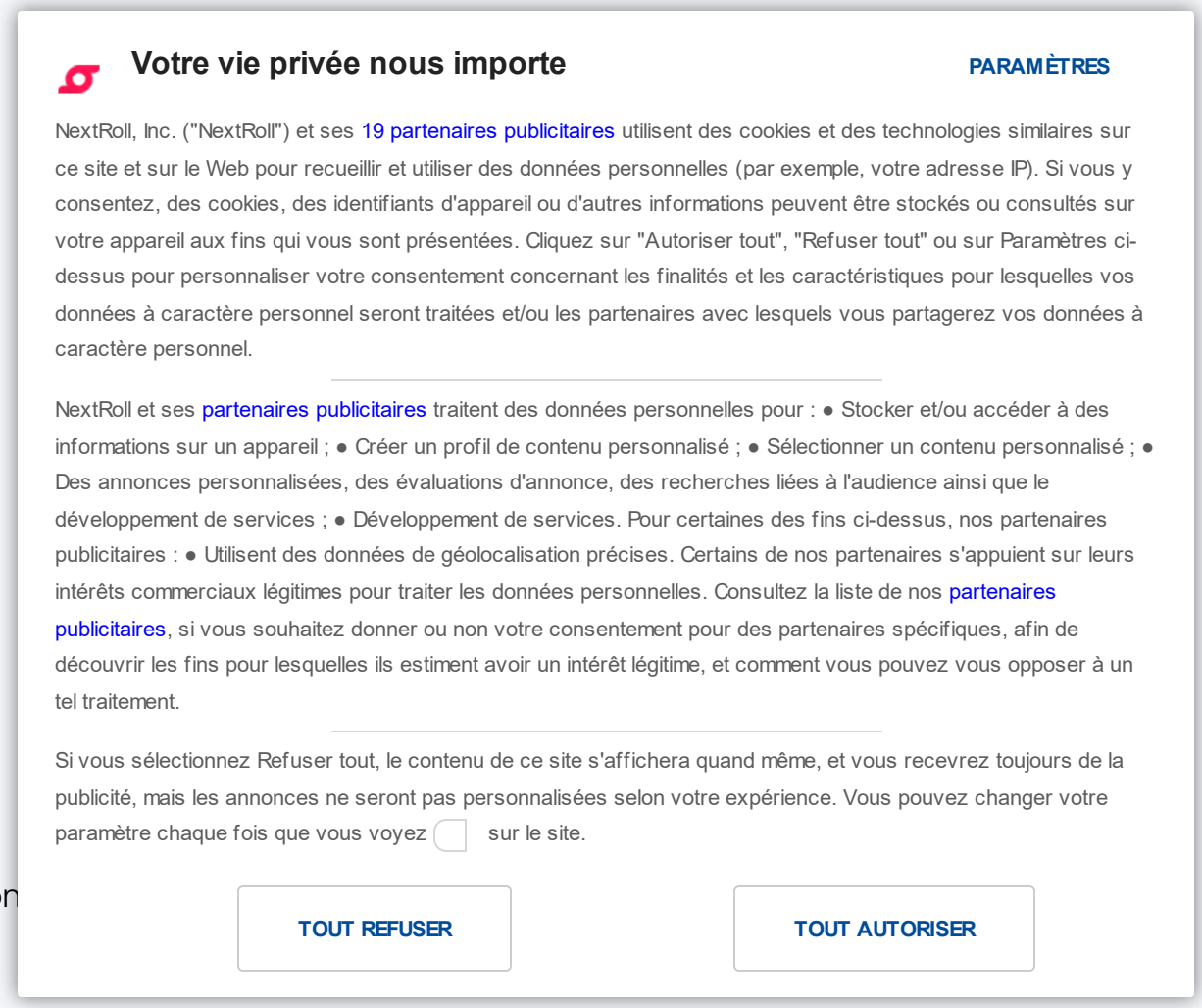


Figure 10: Steals data from chron

While stealing the data from browsers, the malware also checks if keywords belonging to services such as Banking, Cryptocurrency, and Porn are present in the browser data using *ScanData()* method. The Figure below shows the services for which malware runs string search operations.

Figure 11: Checks for specific services

**MS Edge Browsers:**

The malware first checks for the directory "\AppData\Local\Microsoft\Edge\User Data," which helps identify if an edge browser is installed on the victim's system. After this, it enumerates all the files in the system and checks if the "Login Data" file is present. If so, then it steals the data from the browser, as can be seen in the Figure below. Finally, the *ScanData()* method is used again to steal the data from the Edge browser

Figure 12: Steals data from

**Firefox-based browsers:**

Prynt stealer targets eight Firefox-based browsers wh

Figure 13: Targeted Firefox-based browsers

The malware only proceeds to steal data if the Profile folder is present under the "AppData\Browser_name" directory. Firefox Browser uses this folder for saving user data. The malware copies the "logins.json" file from the "Profile" folder to the initially created folder for saving stolen data. The "Logins.json" file is used for storing the Firefox login credentials. Following files are targeted by malware for stealing data, present under the "Profile" folder:

- Places.sqlite (for Bookmarks and History)
- cookies.sqlite (for browser cookies)
- logins.json (for Login Credentials)

Figure 14: Steals data from Firefox-based browsers

## Messaging Applications

After stealing data from browsers, the malware targets the following messaging applications:

- Discord
- Pidgin
- Telegram

The malware first creates a folder names Messenger which will be used for saving data from these applications.

### Discord:

After this, the malware checks for Discord tokens. It fi

- *Discord\\Local Storage\\leveldb*
- *discordptb\\Local Storage\\leveldb*
- *Discord Canary\\leveldb*

It only proceeds if the above directory exists. If direct ending with .ldb or .log and extracts Discord tokens fr creates a folder named "Discord" and will write the st

Figure 15: Steals Discord tokens

## Pidgin:

Pidgin is a chat program that lets you log in to accounts on multiple chat networks simultaneously. It is compatible with the following chat networks: Jabber/XMPP, Bonjour, Gadu-Gadu, IRC, Novell GroupWise Messenger, Lotus Sametime, SILC, SIMPLE, and Zephyr.

The malware first identifies if ".purple\\accounts.xml" is present in the AppData folder. This file stores the Pidgin login credentials. It steals the Login credentials and Protocol details and saves them into the accounts.txt file for exfiltration.

Figure 16: Steals data

## Telegram:

The malware calls Process.GetProcessByName() me
name and path in the victims' machine. The malware
present in the retrieved path. Finally, it gets the Teleg
if it is present—the malware targets "tdata" folder for

Figure 17: Steals telegr

## Gaming Applications

Prynt Stealer targets the following gaming applicati

- Steam
- Minecraft
- Uplay

### Steam:

The malware identifies the Steam installation path by checking the registry key value at "HKEY_LOCAL_MACHINE\Software\Valve\Steam." After this action, it enumerates the subkey present under "HKEY_LOCAL_MACHINE\Software\Valve\Steam\Apps" to get details of the application, as can be seen in the Figure below. The malware also targets the steam's SSFN file, known as the authorization file, and copies it for exfiltration.

Figure 18: Steals data from steam

### Uplay:

The malware looks for "Ubisoft Game Launcher" in the AppData folder, and if this folder is present, it copies all the files in it for exfiltration.

Figure 19: Steals data

### Minecraft:

For Minecraft, the stealer checks if the ".minecraft" fol
directory. If it is present, it creates a folder named "Mi
save the stolen data.

This stealer copies "launcher_profiles.json", "servers.d
for exfiltration. It also extracts mods and version deta
created in "Minecraft" folder.

Figure 20: Steals data from Minecraft

## Crypto Wallets

The malware targets the following crypto wallets:

Zcash, Armory, Bytecoin, Jaxx, Ethereum, AtomicWallet, Guarda,  and Coinomi.

It creates a folder named “Wallets” and then enumerates a list of hardcoded wallets for identifying the crypto wallet used by the victim.

Stealer queries registry for identifying the location of Blockchains such as Litecoin, Dash, and Bitcoin as shown in Figure below. It obtains the path from registry data “strDataDir” in the *HKEY_CURRENT_USER\Software\Blockchain_name\ Blockchain_name-Qt* registry key.

Figure 21: Steals data from

## FTP Applications

Prynt stealer targets FileZilla, a free and open-source
the data from “sitemanager.xml” and “recentservers.
file under the “FileZilla” folder for exfiltration.

Figure 22: Steals data from FileZilla

## VPN

Prynt Stealer targets the following VPN applications:

- OpenVPN
- PorotonVPN
- NordVPN

It copies the configuration file of ProtonVPN, OpenVPN and steals the user credentials from NordVPN configuration file.

Figure 23: Steals data from VPN's configuration file

## Directory tree

After this action, the malware creates a folder named "Directories" and then obtains the structure of a directory and writes them to text files, as shown in the Figure below. The directories targeted by malware include the one targeted initially for copying data.



Figure 24: Obtains the d

## System Information

It creates a folder named "System" in which it will sto
running processes, network details, and victim's syste

**Process Details:**

Prynt stealer uses *Process.GetProcesses()* method to
victim's system and write them to the "Process.txt" file

- Process name
- Process ID
- Executable path

After this action, it gets the active windows using the *process.MainWindowTitle*() method and write the data into the “Windows.txt” file in the format:

- Process name
- Process ID
- Executable path

Figure 25: Extract details of current processes

**Screenshot:**

Now it takes a screenshot of the victim’s system and saves it as a “Desktop.jpg” file:

Figure 26: Takes Screenshot

**Network Information:**

The stealer also extracts the network credentials usin
*wlan show profile”* and saves them into the “Savedne
command *“/C chcp 65001 && netsh wlan show netw*
available networks and saves them into the “Scannin

Figure 27: Steals save network credentials and identify the available network

### Windows Product Key:

It steals the windows product key from the "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion," decodes it, and then saves it to the "ProductKey.txt file."

Figure 28: Steal Windows product key

## Data exfiltration:

The malware creates a list and adds the overview of
below. Then it sends a chat message using the Teleg

For identifying the public IP, it sends a request to hxxp

For identifying the geolocation, it sends a request to
hxxps[:]//api.myInikov.org/geolocation/wifi?v=1.1&bss

Figure 29: Creates an overview of stolen data

The malware compresses the folder where the stolen data is saved and exfiltrates it to the telegram bot. Furthermore, it uses a secure network connection for exfiltrating the stolen data to the remote server.

Figure 30: Decrypted ne

## Other Capabilities

Our analysis found that specific modules in the samp
including the Anti-analysis, Keylogger, and Clipper. T
for this stealer, which can be customized to control th
anti-analysis, it's working on the hardcoded string pr
the method responsible for executing anti-analysis fu
also depend on these hard-coded strings.

Figure 31: Anti-analysis

## Clipper:

The Figure below shows the list in which TAs can store their crypto addresses. These entries are not populated, highlighting the fact that TA might not have opted for this functionality in the builder.

Figure 32: Clipper

## Keylogger:

This stealer enables the keylogging feature only if the
running in the system. The stolen data will be saved i

Figure 33: Keylogger module

## Conclusion

Prynt Stealer is a recent infostealer strain. It has a ton of capabilities. Though there are pretty popular stealers in the cybercrime marketplaces, TAs do adopt new toolkits which aid them in updating their Tactics, Techniques, and Procedures. These types of malware provide an easy way for TAs to get into the corporate networks, as breaking into one network is not everyone's cup of tea.

## Our Recommendations:

- Avoid downloading pirated software from warez/torrent websites. The "Hack Tool" present on sites such as YouTube, torrent sites, etc., mainly 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/family members to prevent successful phishing's/untrusted URLs.
- Block URLs that could be used to spread the m
- Monitor the beacon on the network level to blo
- Enable Data Loss Prevention (DLP) Solution on t

## MITRE ATT&CK® Techniques

Previous
Fake MetaMask App Steals Cryptocurrency

Next
Malicious App Targets Turkish Ministry Of Justice

## Related Posts

IT Vulnerability Report: Fortinet, SonicWall Hac
Exposures Top 1 Million
November 1, 2024

## Indicators of Compromise (IoCs):

| Indicators | Indicator type | I |
|---|---|---|
| ab913c26832cd6e038625e30ebd38ec2<br>719873f61eeb769493ac17d61603a6023a3db6dd<br>1283c477e094db7af7d912ba115c77c96223208c03841768378a10d1819422f2 | **MD5**<br>**SHA1**<br>**SHA256** | |
| 0b75113f8a78dcc1dea18d0e9aabc10a<br>269e61eed692911c3a886a108374e2a6d155c8d1<br>808385d902d8472046e5899237e965d8087da09d623149ba38b3814659689906 | **MD5**<br>**SHA1**<br>**SHA256** | |
| 661842995f7fdd2e61667dbc2f019ff3<br>1a638a81b9135340bc7d1f5e7eae5f3f06667a42<br>4569670aca0cc480903b07c7026544e7e15b3f293e7c1533273c90153c46cc87 | **MD5**<br>**SHA1**<br>**SHA256** | |