Mint Stealer: A Comprehensive Study of a Python-Based Information Stealer - CYFIRMA

Archived: 2026-04-06 00:38:38 UTC

Published On : 2024-07-30

Executive Summary

At Cyfirma, we are dedicated to provide current insights into prevalent threats and the strategies employed by malicious entities targeting both organizations and individuals. This report offers a comprehensive analysis of Mint Stealer, an information-stealing malware operating within a malware-as-a-service (MaaS) framework. Mint Stealer is designed to target sensitive data and uses sophisticated techniques to evade detection. This report explores Mint Stealer’s evasion tactics, methods for concealing malicious activities, and highlights the evolving strategies of cyber threat actors in the contemporary threat landscape.

Introduction

Mint-stealer is a potent piece of malware operating as a malware-as-a-service (MaaS) tool, designed to covertly exfiltrate a wide range of sensitive data from compromised systems. This malware targets and extracts critical

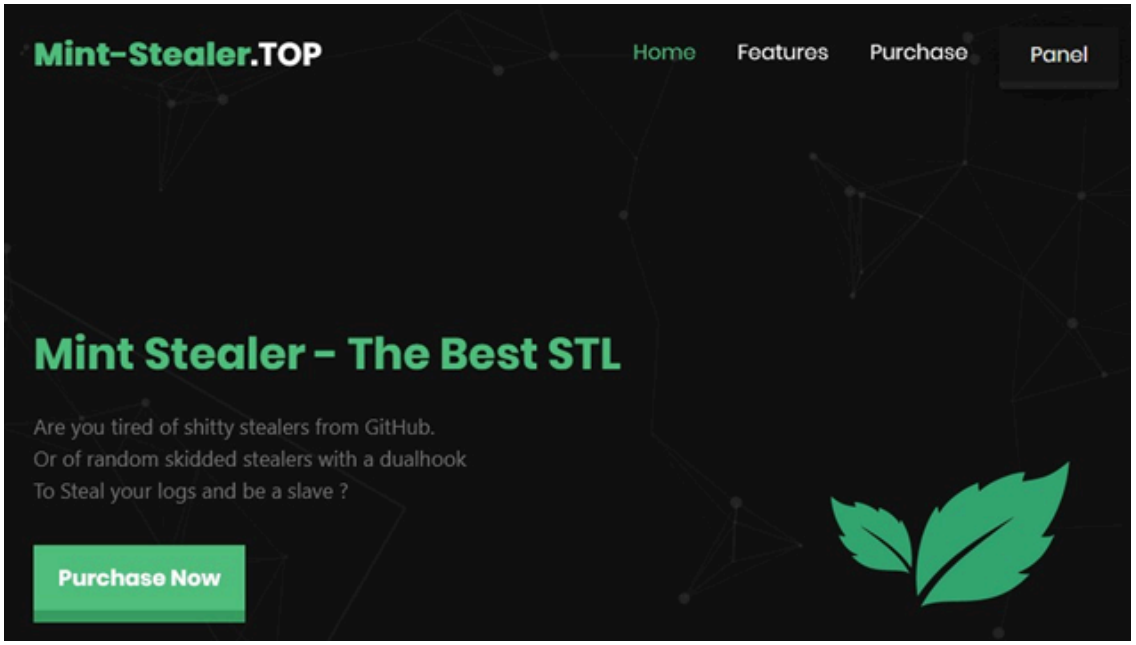
information, including web browser data, cryptocurrency wallet details, gaming credentials, VPN client information, messaging app data, FTP client data, and more.
Mint-stealer employs techniques such as encryption and obfuscation to evade detection and enhance its effectiveness. It is marketed and sold through multiple dedicated websites, with support provided through Telegram. This report examines Mint-stealer's operational methods, its impact on cybersecurity, and offers guidance for professionals on developing effective defense strategies against such sophisticated threats.

Key Findings

- Mint-stealer is a potent malware functioning as a malware-as-a-service (MaaS) tool, designed to covertly exfiltrate a wide range of sensitive data from compromised systems.
- It targets data from web browsers, cryptocurrency wallets, gaming credentials, VPN clients, messaging apps, and FTP client data.
- This malware is sold through multiple dedicated websites, with support provided via Telegram.
- It is also associated with other malware-selling sites and the hosting service that facilitates malicious activities.
- Mint-stealer is created using the Nuitka Python compiler and relies on Python dynamic modules to support its functionality.
- The primary specimen acts as a dropper, with the main payload hidden in a compressed form within the resource section of the executable.
- Checks for debuggers and analysis tools running in the environment.
- Mint-stealer uploads stolen data to free file-sharing websites and then sends the URL of the uploaded data to its command-and-control server (C2).
- It sends and receives updates and instructions from the C2 server.

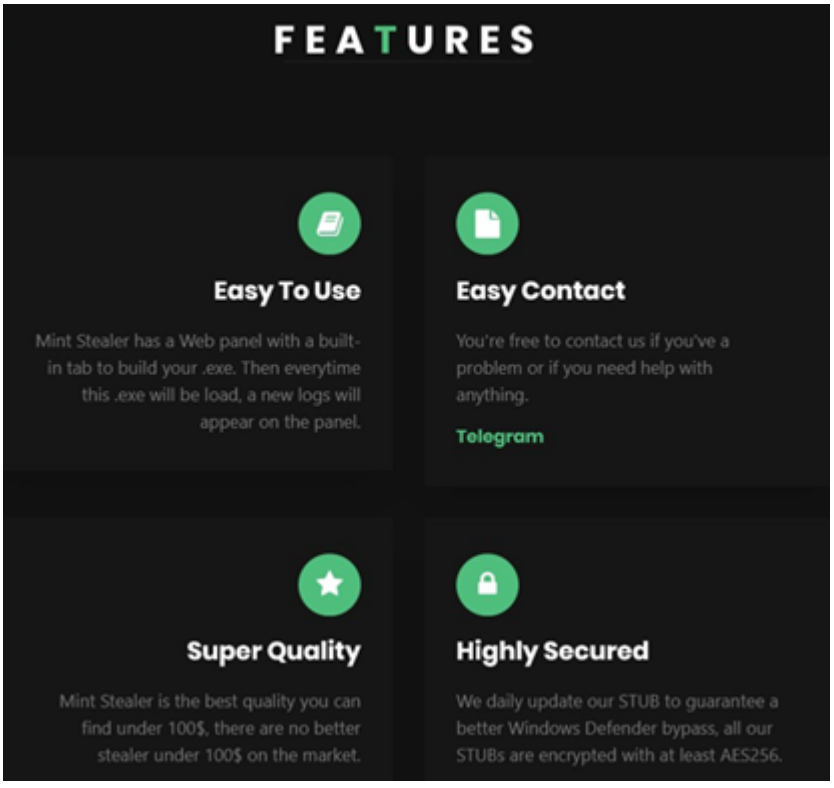
ETLM Attribution

Mint-stealer is being sold as malware-as-a-service (MaaS)on mint-stealer[.]top and mint-c2[.]top, with both domains hosting the same website. Additionally, mint-c2[.]top is used as the command and control (C2) server for the stealer:

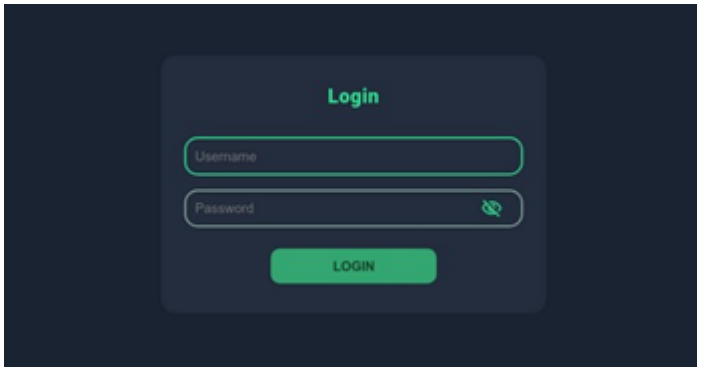


mint-c2[.]top/ mint-c2[.]top

The website details Mint-stealer's features, including its usage, Telegram contact support, and frequent updates to bypass Windows Defender. It also claims to be the best stealer available at a low price:

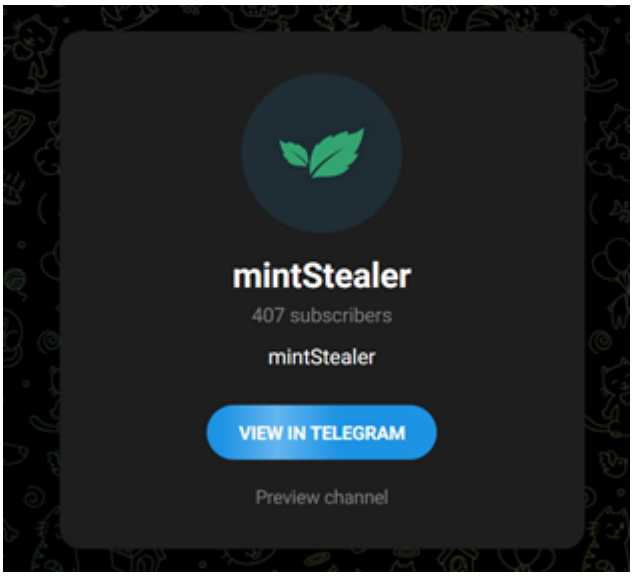


Mint-stealer provides a login panel for its subscribers to access the stealer logs from compromised systems:

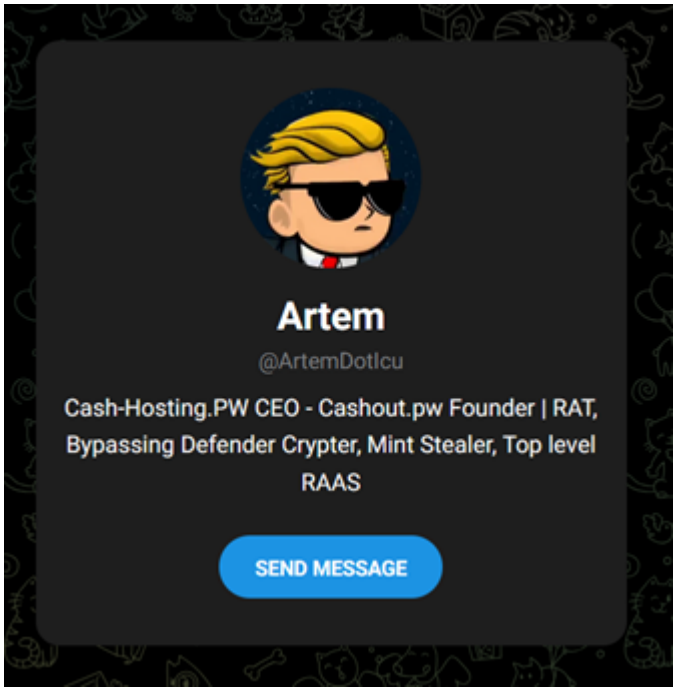


Stealer login panel: mint-stealer[.]top/panel/login

It also provides support through a Telegram group and has 407 subscribers at the time of writing:

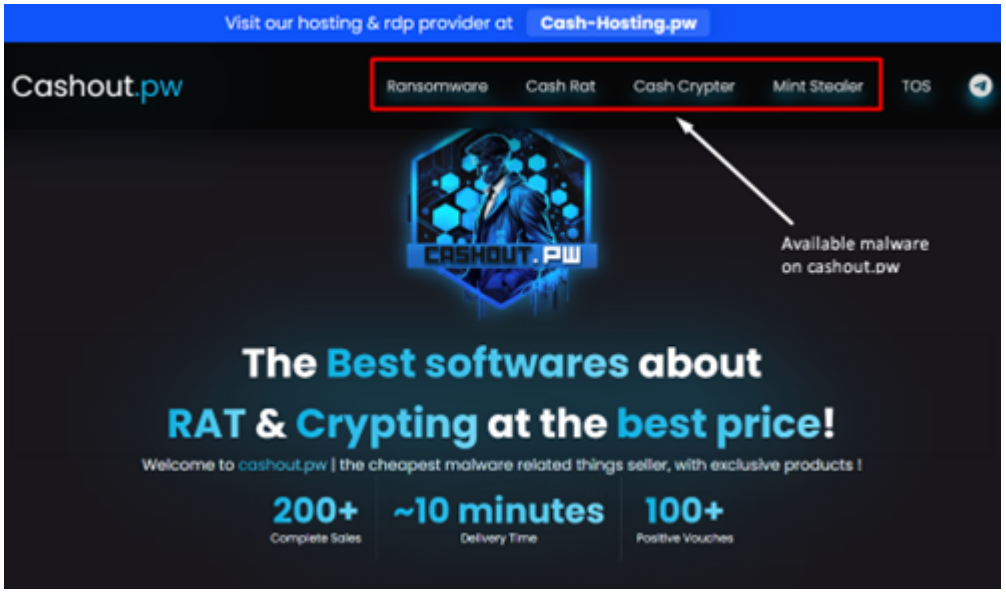


The threat actor behind the Mint-stealer has provided a Telegram contact on their website, confirming their association with another malware-selling website, cashout[.]pw:



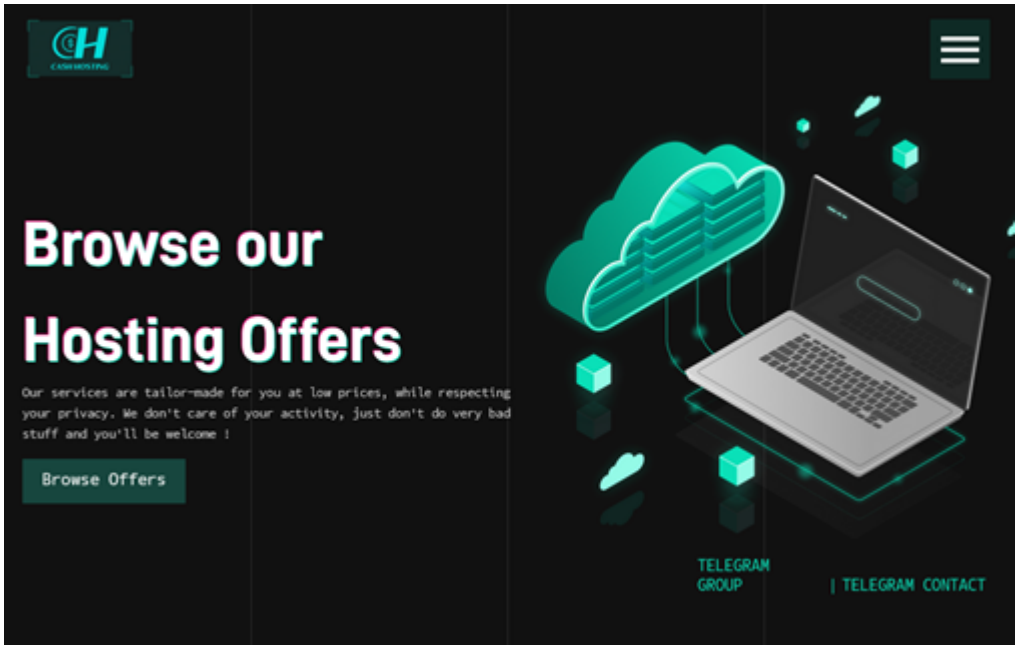
Threat actor's telegram contact

cashout[.]pw sells malware including RATs (Remote Access Trojans), crypters, and ransomware, and also features Mint-stealer:



Cashout[.]pw

The threat actor(s) also offer hosting services on cash-hosting[.]pw, including VPN, VPS, RDP, and cPanel, which do not respect DMCA requests:



cash-hosting[.]pw

Threat Landscape:

The external threat landscape is constantly shifting, with sophisticated threats like Mint-stealer emerging as new challenges. The creator(s) of Mint-stealer are particularly skilled at adapting their tactics, using methods like encryption and obfuscation to stay under the radar and strengthen their attacks. Their methods also involve utilizing unrestricted hosting services and maintaining robust command and control systems. This ongoing adaptability highlights the persistent challenge in cybersecurity and highlights the necessity for continuous vigilance and evolving defense strategies.

Analysis of Mint Stealer

File Analysis	
File Name	Setup.exe
File Size	9.49 MB (9955840 bytes)
Signed	Not signed
MD5	e6e620e5cac01f73d0243dc9cf684193
SHA-256	1064ab9e734628e74c580c5aba71e4660ee3ed68db71f6aa81e30f148a5080fa
Date Modified	23-06-2024

The primary specimen of Mint-stealer is a 64-bit console-based executable, compiled using Microsoft Visual C/C++. It is originally named vadimloader.exe and claims to be a Setup file copyrighted by Microsoft:

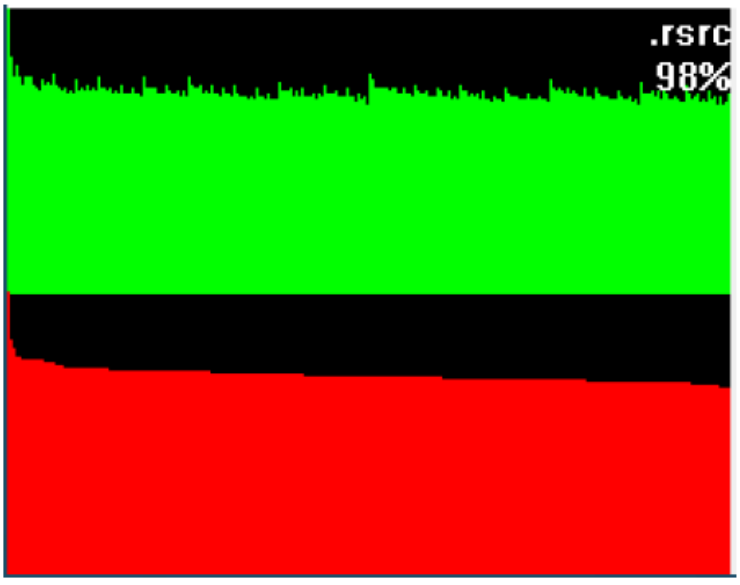
PE64
Compiler: Microsoft Visual C/C++(2015 v.14.0)[-]
Linker: Microsoft Linker(14.0)[Console64,console]

language	neutral
code-page	Unicode UTF-16, little endian
CompanyName	Microsoft
ProductName	Setup
FileDescription	Setup
LegalCopyright	Copyright Microsoft Corporation. All rights reserved.
ProductVersion	1.1.0.0
FileVersion	1.1.0.0
OriginalFilename	**vadimloader.exe**
InternalName	vadimloader

The executable consists of 9 sections, with the resource section containing 97.51% of the file data and an entropy of 7.999. The green region of the byte-usage-histogram shows all possible byte values (ranging from 0x00 to 0xFF) on the X-axis, with their frequency of occurrence on the Y-axis. The red region of the histogram orders these byte values in descending order of occurrence. The section's high entropy and the uniform distribution of byte values confirm that the resource section is compressed:

section[7]
.rsrc
5B07CDC02DDFDB77B32AF1...
7.999
97.51 %
0x0003C000
0x0097E200
0x00942200 (9708032 bytes)
0x00054000
0x00942138 (9707832 bytes)

Resource section data



byte-usage-histogram

The specimen does not require administrative rights and can execute with the current user's privileges:

```
<ns2:security>
  <ns2:requestedPrivileges>
    <ns2:requestedExecutionLevel level="asInvoker" uiAccess="false"/>
  </ns2:requestedPrivileges>
```

Behavioral & Code Analysis

1st Stage Execution:

In the initial stage of execution, Setup.exe accesses its resource section to retrieve the content that it will use as the next stage payload:

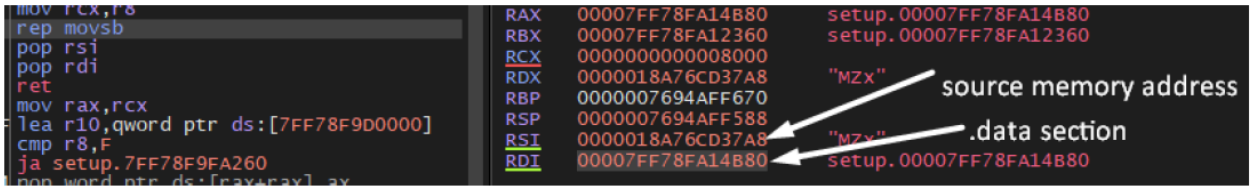
```
mov     edx, 1Bh        ; lpName
mov     r8d, 0Ah        ; lpType
xor     ecx, ecx        ; hModule
call    cs:FindResourceA
mov     rdi, rax
xor     ecx, ecx        ; hModule
mov     rdx, rax        ; hResInfo
call    cs:LoadResource
mov     rcx, rax        ; hResData
call    cs:LockResource
mov     cs:qword_140042358, rax
mov     cs:qword_14003FB10, rax
xor     ecx, ecx        ; hModule
mov     rdx, rdi        ; hResInfo
call    cs:SizeofResource
mov     eax, eax
mov     cs:qword_14003FB30, rax
mov     rdx, cs:qword_14003FB10
movzx   r8d, byte ptr [rdx]
movzx   ecx, byte ptr [rdx+1]
movzx   eax, byte ptr [rdx+2]
add     rdx, 3
mov     cs:qword_14003FB10, rdx
cmp     r8b, 4Bh ; 'K'
jnz     loc_1400135B4
```

Setup.exe creates a directory under the user's Temp directory (C:\Users\user\AppData\Local\Temp). The directory name is based on the string 'onefile,' the process ID of Setup.exe, and the system time (retrieved using the GetSystemTimeAsFileTime API):

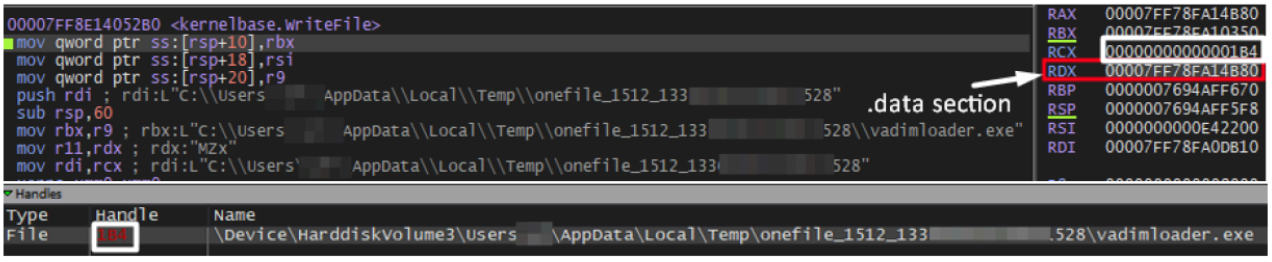
```
lea     r15, FileName
lea     rdx, aTempOnefilePid ; "{TEMP}\\onefile_{PID}_{TIME}"
mov     r8d, 1000h
mov     rcx, r15        ; String
call    sub_1400123E0
```

Setup.exe creates a directory in the Temp folder

Next, it creates a file named vadimloader.exe inside the newly created directory. Then, it writes the executable code from memory (loaded earlier from the resource section) to the .data section, which is subsequently written to vadimloader.exe:

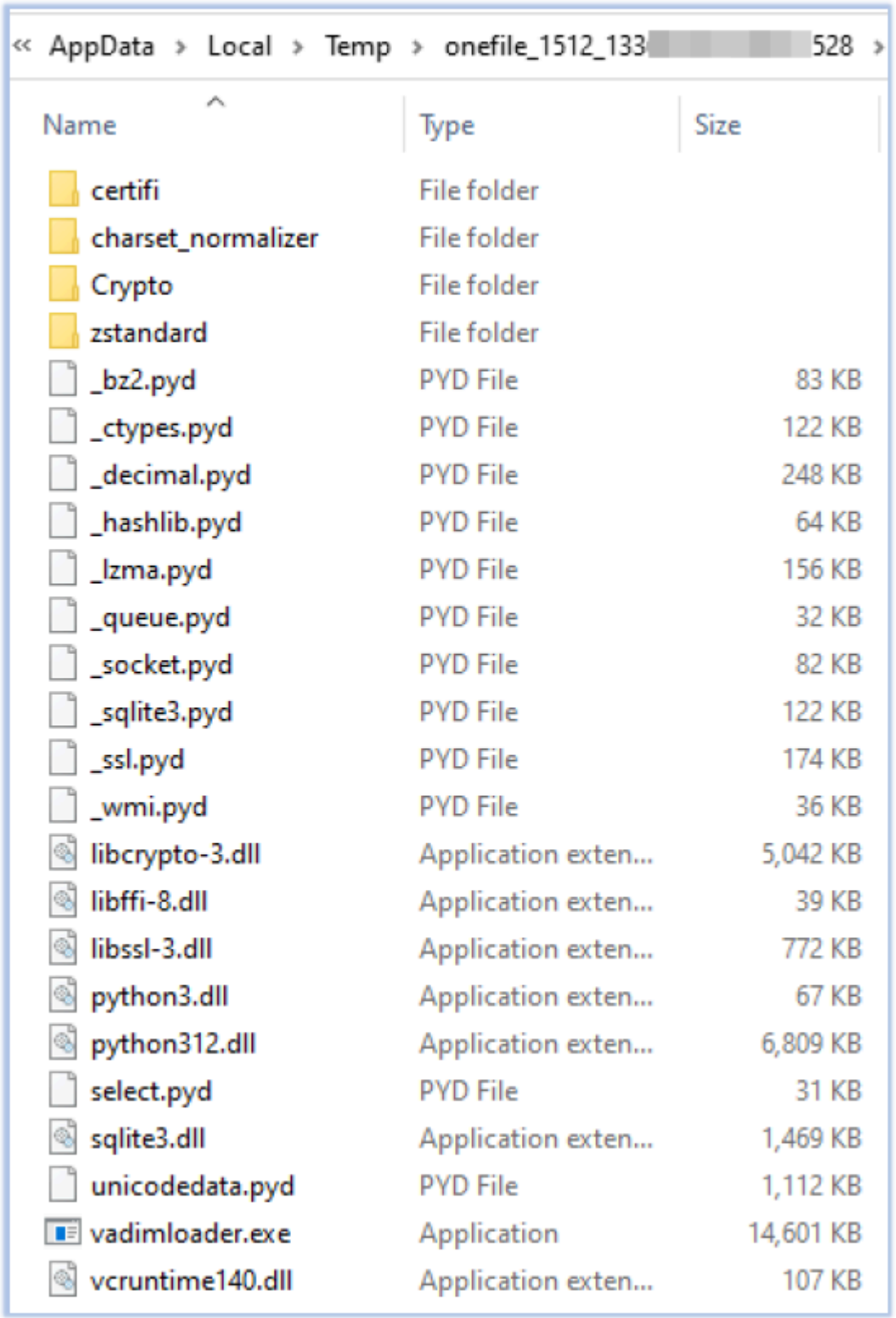


Writing to .data section from process memory



Writing to *vadimloader.exe* from .data section

In a similar manner, it also drops additional files into the same directory (Temp/onefile_1512_…). These files include Python dynamic modules (‘.pyd’ files), DLLs, and a file containing CA certificates (‘cacert.pem’):



Dropped files in *Temp/onefile_1512_...*

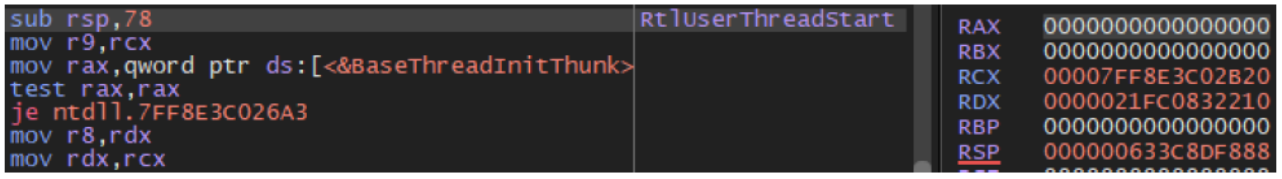
File Name	vadimloader.exe
File Size	14.26 MB (14950912 bytes)

Signed	Not signed
MD5	9f037593071344bc1354e5a619f914f4
SHA-256	db47e673cccdbe2abb11cc07997aeabf4d2bdc9bec286674b58c6baafa09b823
Date Modified	23-06-2024

vadimloader.exe is a 64-bit console-based Windows executable, created using the Nuitka Python compiler. The Temp/onefile_1512_… directory contains all the files needed to support the execution and functionality of the Mint-stealer.

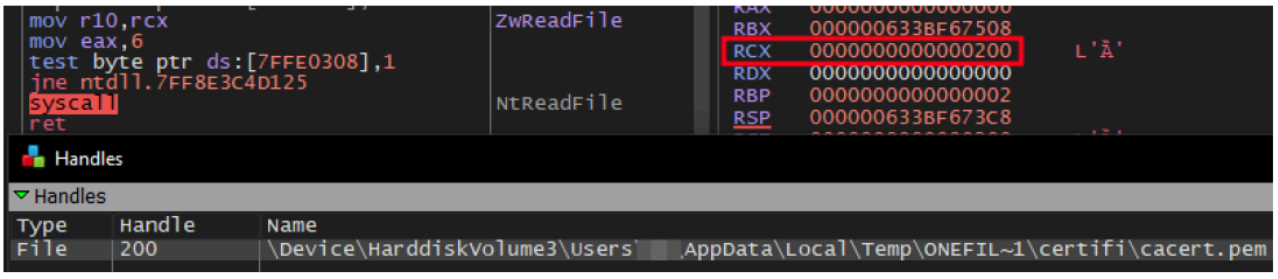
2nd Stage Execution:

In the second stage, Setup.exe executes vadimloader.exe as a child process, using the RtlUserThreadStart API call. This function does not require explicit definition or adjustment of the thread context before resuming execution. It sets up the initial context and prepares the thread to execute user-defined code, and the Windows kernel ensures that the thread's initial context, including registers and stack setup, is properly initialized for execution.



Resuming *vadimloader.exe* execution

vadimloader.exe reads all the files in the Temp/onefile_1512_… directory, including subfolders, and loads the required libraries and code into the process memory for its operation.

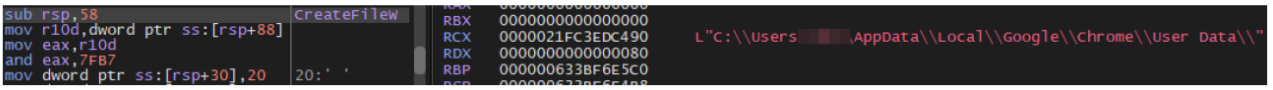


Reading CA Certificates from *cacert.pem*

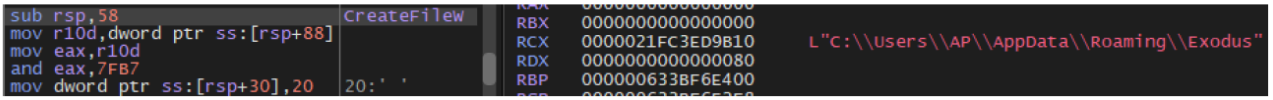
3rd Stage Execution:

Mint-stealer begins collecting data from the infected system, including web browser data, cryptocurrency wallet information, gaming data, VPN client details, messaging applications, FTP clients, file management applications, and clipboard data.

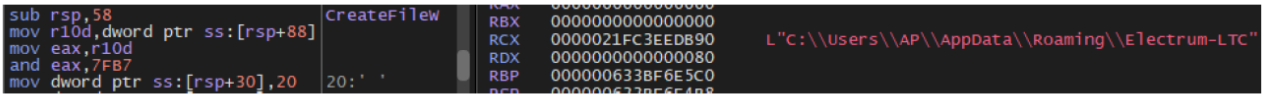
Collecting Brave browser user data



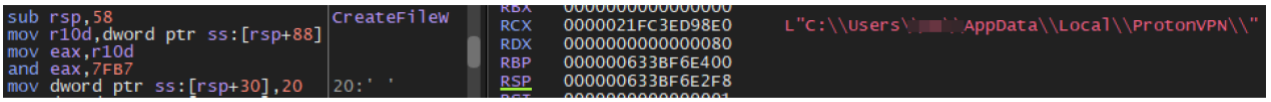
Collecting Google Chrome browser user data



Collecting Exodus wallet data



Collecting Electrum wallet data



Collecting Proton VPN data

The following applications and services are targeted by the Mint-stealer:

•**Web Browsers:** Opera, Edge, Mozilla Firefox, Yandex, Iridium, Epic, Sputnik, 7star, Cent, Orbitum, Kometa, Torch, Amigo, Thunderbird, Vivaldi.

•**Cryptocurrency Wallets:** Exodus, Electrum, Atomic, MultiDoge, Bitcoin Core, Binance, Coinomi, Jaxx, Electron Cash, Ethereum

•**Gaming:** Battle.net, Growtopia, Minecraft, Purple

•**VPNs:** Proton VPN, OpenVPN

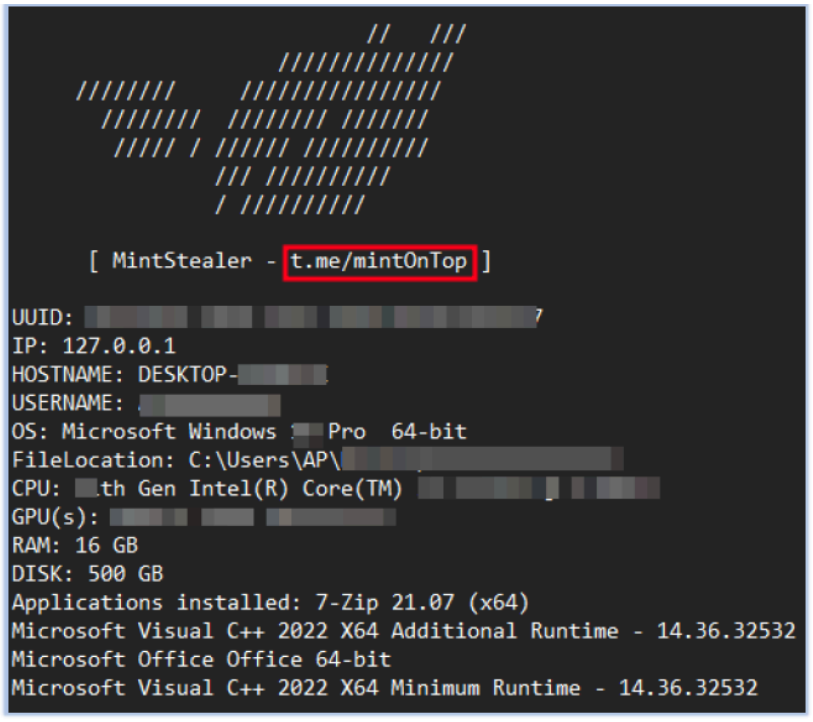
•**Messaging/Chat Applications:** Skype, Element, Signal, ICQ, Steam, Telegram, Tox.

•**FTP/File Management:** FileZilla, Shadow (PC & Drive), Ghisler Total Commander

Mint-stealer also collects system information using wmic commands:

hostname.exe (5340)	hostname
cmd.exe (3560)	C:\Windows\system32\cmd.exe /c "wmic csproduct get uuid | more +1"
WMIC.exe (9000)	wmic csproduct get uuid
more.com (8564)	more +1
cmd.exe (4140)	C:\Windows\system32\cmd.exe /c "wmic OS get caption, osarchitecture | more +1"
WMIC.exe (8448)	wmic OS get caption, osarchitecture
more.com (7536)	more +1
cmd.exe (2560)	C:\Windows\system32\cmd.exe /c "wmic cpu get name | more +1"
WMIC.exe (2020)	wmic cpu get name
more.com (2312)	more +1
cmd.exe (1812)	C:\Windows\system32\cmd.exe /c "wmic PATH Win32_VideoController get name | more +1"
WMIC.exe (2480)	wmic PATH Win32_VideoController get name
more.com (1712)	more +1
cmd.exe (1448)	C:\Windows\system32\cmd.exe /c "wmic computersystem get totalphysicalmemory | more +1"
WMIC.exe (8652)	wmic computersystem get totalphysicalmemory
more.com (1036)	more +1

Process tree: system information gathering



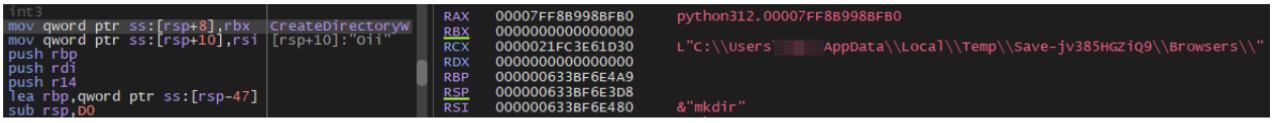
System information saved in *Temp/onefile_1512_.../Save-~/information.txt* file

It also continuously executes PowerShell commands to capture clipboard data:

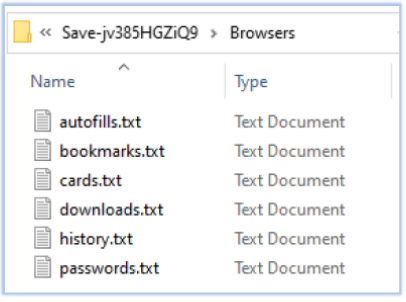
powershell.exe (2176)	powershell Get-Clipboard
powershell.exe (5856)	powershell Get-Clipboard
powershell.exe (5160)	powershell Get-Clipboard
powershell.exe (6016)	powershell Get-Clipboard
powershell.exe (4876)	powershell Get-Clipboard
powershell.exe (804)	powershell Get-Clipboard
powershell.exe (3084)	powershell Get-Clipboard
powershell.exe (8536)	powershell Get-Clipboard
powershell.exe (6536)	powershell Get-Clipboard
powershell.exe (9048)	powershell Get-Clipboard
powershell.exe (8120)	powershell Get-Clipboard
powershell.exe (6504)	powershell Get-Clipboard
powershell.exe (8848)	powershell Get-Clipboard
powershell.exe (8156)	powershell Get-Clipboard
powershell.exe (8356)	powershell Get-Clipboard
powershell.exe (2060)	powershell Get-Clipboard
powershell.exe (8092)	powershell Get-Clipboard
powershell.exe (8312)	powershell Get-Clipboard

PowerShell commands capturing clipboard data

The malware creates another directory within Temp/onefile_1512_… named ‘Save-’ followed by a randomly generated string, and saves all the harvested data into that directory

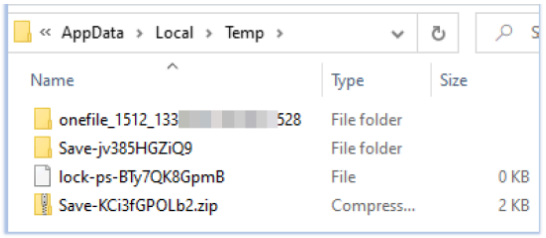


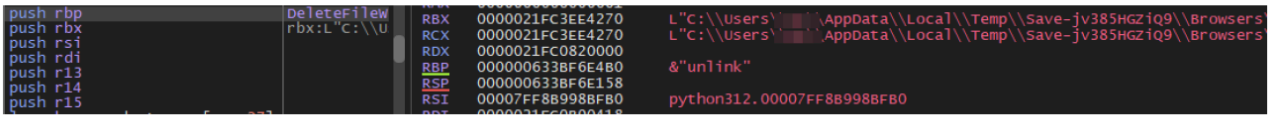
Creating a *Browsers* directory to save harvested browser data



Captured browser data

The Save-~ folder is then compressed into a ZIP archive with a name starting with ‘Save-’ followed by a different random string before being deleted:



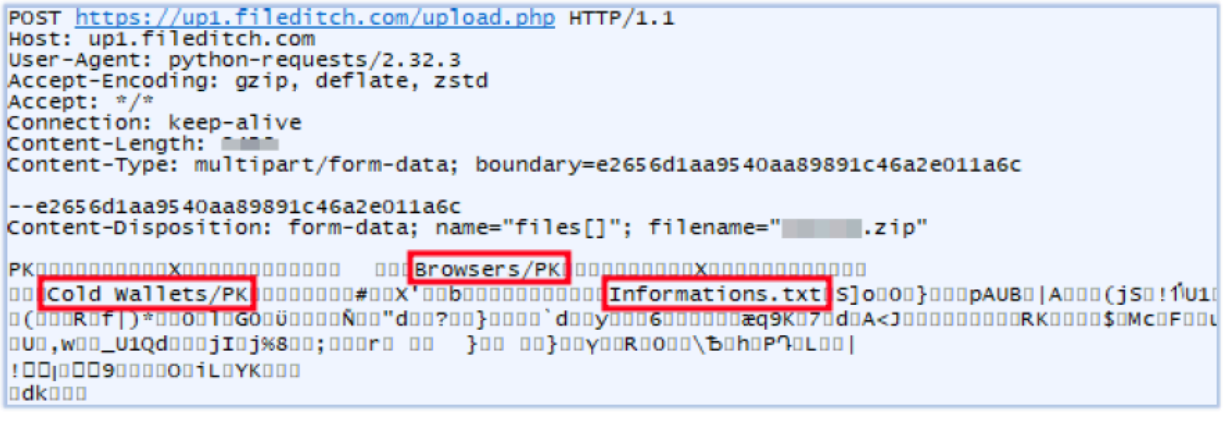


Deleting harvested browser data from the *Save-~* folder

The Exfiltration:

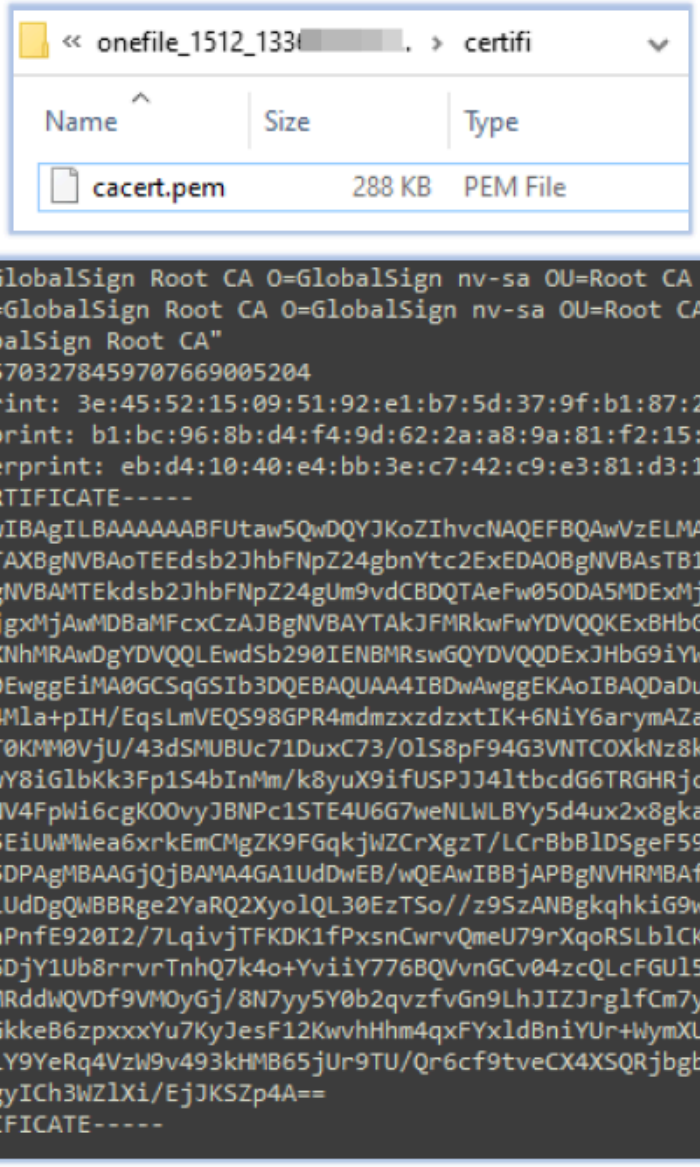
The malware first checks the IP address of the compromised host by sending an HTTP request to api[.]ipify[.]org, and then uploads the ZIP file (containing harvested data) to free file hosting sites, such as anonfiles[.]com, gofile[.]io, and fileditch[.]com:

Uploading data to gofile[.]io



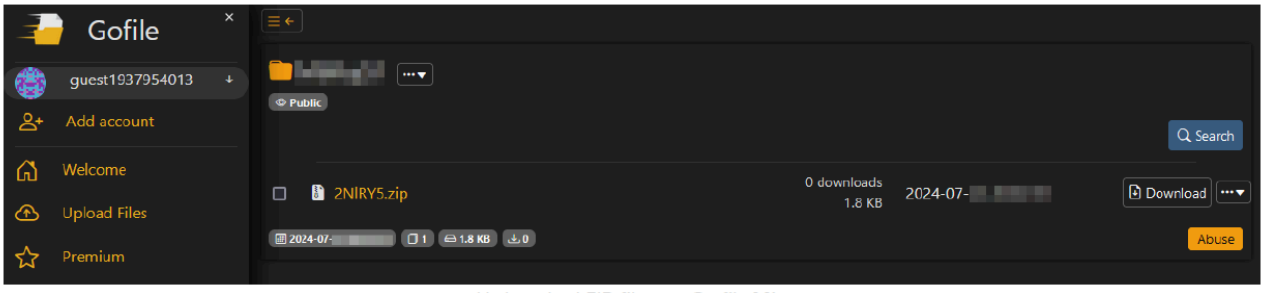
Uploading data to fileditch[.]com

The cacert.pem file (Temp/onefile_1512_…/certifi/cacert.pem) contains multiple CA certificates, which Mint-stealer uses to encrypt data over the network:



A Snippet of certificates stored in *cacert.pem* file

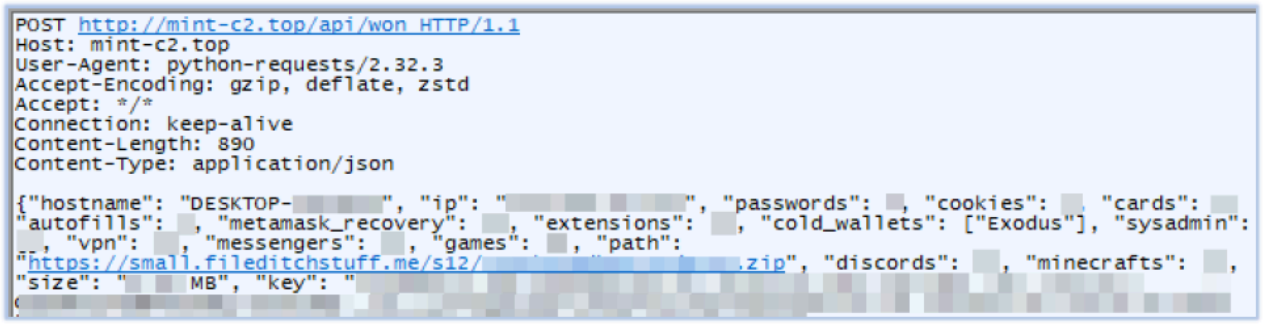
The file upload request responds with a specific URL from which the uploaded file can be downloaded, and it is publicly accessible:



Uploaded ZIP file on *Gofile[.]io*

Interestingly, the Mint-stealer sends a summary of the exfiltrated data, which has already been uploaded to the file-sharing site, along with the URL for downloading the file, to its C2 server (mint-c2[.]top). This transmission occurs over an unsecured network, unlike the file upload, which is done over a secure connection:

C2 communication: summary of exfiltrated data on *gofile[.]io*



C2 communication: summary of exfiltrated data on *fileditch[.com]*

Mint-Stealer Capabilities

Analyzing Mint-stealer offers important insights into its operational features. Based on this analysis, the following points highlight the capabilities of this information-stealing malware:

1.Targets and steals a wide range of sensitive information, including web browser data, cryptocurrency wallet details, gaming credentials, VPN client information, messaging app data, and FTP client data.

2.Captures system information.

3.Creates and manages directories in the TEMP folder to store and organize the harvested data.

4.Detects debugger and analysis environment.

5.Continuously captures clipboard data through PowerShell commands.

6.Encrypts exfiltrated data to enhance security and evade detection during unauthorized data transfers.

7.It sends and receives updates and instructions from the C2 server.

Conclusion:

The examination of the Mint-stealer reveals a sophisticated and versatile information-stealing malware that operates as a malware-as-a-service (MaaS) tool. It effectively exfiltrates a wide array of sensitive data from compromised systems, including web browser information, cryptocurrency wallet details, and more. By leveraging advanced techniques, such as encryption, obfuscation, and file compression, Mint-stealer evades detection and maximizes its impact. Its operational model, involving data uploads to free file-sharing sites and communications with command-and-control servers, underscores its adaptability and the significant threat it poses. The malware's distribution through specialized websites and support via Telegram highlights the broader ecosystem of cybercrime, emphasizing the need for robust and adaptive cybersecurity measures to counter such evolving threats.

As threats like Mint-stealer continue to evolve, it is important for organizations to implement robust cybersecurity measures and proactive defense strategies to mitigate the associated risks. To reduce the threat of Mint-stealer, users should exercise caution when opening files from untrusted sources or clicking on unfamiliar links, especially those promoting dubious software or content. Additionally, deploying strong cybersecurity practices – such as using reputable antivirus software, keeping all software up to date, and remaining vigilant against social engineering attacks – can significantly enhance protection against such sophisticated malware.

Indicators Of Compromise

Indicators	Type	Context
e6e620e5cac01f73d0243dc9cf684193	File	Setup.exe
1064ab9e734628e74c580c5aba71e4660ee3ed68db71f6aa81e30f148a5080fa	File	Setup.exe
9f037593071344bc1354e5a619f914f4	File	vadimloader.exe
db47e673cccdbe2abb11cc07997aeabf4d2bdc9bec286674b58c6baafa09b823	File	vadimloader.exe
mint-c2[.]top	Domain	C2
mint-stealer[.]top	Domain	C2
mint-c2[.]top/api/won	URL	Exfiltration
mint-c2[.]top/api/injection	URL	Exfiltration
188[.]114[.]96[.]3	IP address	C2
94[.]156[.]79[.]162	IP address	C2
cashout[.]pw	Domain	C2

MITRE ATT&CK Tactics and Techniques

No.	Tactic	Technique
1	Reconnaissance (TA0043)	T1592: Gather Victim Host Information
2	Execution (TA0002)	T1204.002: Malicious File
4	Defense Evasion (TA0005)	T1622: Debugger Evasion
		T1497: Virtualization/Sandbox Evasion
		T1140: Deobfuscate/Decode Files or Information
5	Discovery (TA0007)	T1622: Debugger Evasion

		T1497: Virtualization/Sandbox Evasion
		T1083: File and Directory Discovery
6	Command and Control (TA0011)	T1071.001: Web Protocols
7	Exfiltration (TA0010)	T1041: Exfiltration Over C2 Channel

YARA Rules

```
rule MintStealer
{
meta:
description = “Detects Mint-stealer based on known IoCs”
author = Cyfirma Research
strings:
$setup_exe_hash = “e6e620e5cac01f73d0243dc9cf684193” // MD5 hash of Setup.exe
$setup_exe_hash_alt = “1064ab9e734628e74c580c5aba71e4660ee3ed68db71f6aa81e30f148a5080fa” // SHA-256 hash of Setup.exe
$vadimloader_exe_hash = “9f037593071344bc1354e5a619f914f4” // MD5 hash of vadimloader.exe
$vadimloader_exe_hash_alt = “db47e673cccdbe2abb11cc07997aeabf4d2bdc9bec286674b58c6baafa09b823” // SHA-256 hash of vadimloader.exe
$c2_domain1 = “mint-c2.top”
$c2_domain2 = “mint-stealer.top”
$url1 = “mint-c2.top/api/won”
$url2 = “mint-c2.top/api/injection”
$ip_address1 = “188.114.96.3”
$ip_address2 = “94.156.79.162”
$malware_site = “cashout.pw”
condition:
(any of ($setup_exe_hash, $setup_exe_hash_alt) or
any of ($vadimloader_exe_hash, $vadimloader_exe_hash_alt)) or
(any of ($c2_domain1, $c2_domain2) or
any of ($url1, $url2) or
any of ($ip_address1, $ip_address2) or
$malware_site)
}
```

Recommendations

• Implement threat intelligence to proactively counter the threats associated with the Mint-stealer.

• To protect the endpoints, use robust endpoint security solutions for real-time monitoring and threat detection, such as Antimalware security suit and host-based intrusion prevention system.

• Continuous monitoring of the network activity with NIDS/NIPS and using the web application firewall to

filter/block suspicious activity provides comprehensive protection from compromise due to encrypted payloads.
• Configure firewalls to block outbound communication to known malicious IP addresses and domains associated with Mint-stealer command and control servers.
• Implement behavior-based monitoring to detect unusual activity patterns, such as suspicious processes attempting to make unauthorized network connections.
• Employ application whitelisting to allow only approved applications to run on endpoints, preventing the execution of unauthorized or malicious executables.
• Conducting vulnerability assessment and penetration testing on the environment periodically helps in hardening the security by finding the security loopholes, followed by a remediation process.
• The use of security benchmarks to create baseline security procedures and organizational security policies is also recommended.
• Develop a comprehensive incident response plan that outlines steps to take in case of a malware infection, including isolating affected systems and notifying relevant stakeholders.
• Security awareness and training programs help to protect from security incidents such as social engineering attacks. Organizations should remain vigilant and continuously adapt their defenses to mitigate the evolving threats posed by the Mint-stealer malware.
• Update security patches which can reduce the risk of potential compromise.
