COM Object hijacking: the discreet way of persistence

By Paul Rascagneres

Published: 2016-11-25 · Archived: 2026-04-05 17:27:44 UTC

10/30/2014

Reading time: 3 min (695 words)

G DATA SecurityLabs experts discovered a new Remote Administration Tool, which we dubbed COMpfun. This RAT supports 32-bit and 64-bit Windows versions, up to the Windows 8 operating system. The features are rather common for today's espionage tools: file management (download and upload), screenshot taking, Keylogger functionality, code execution possibility and more. It uses the HTTPS and an asymmetric encryption (RSA) to communicate with the command and control server. The big novelty is the persistence mechanism: the malware hijacks a legitimate COM object in order to be injected into the processes of the compromised system. And it is remarkable, that this hijacking action does not need administrator rights. With this RAT, Attackers could spy on an infected system for quite a long time, as this detection evasion and persistence mechanism is indeed pretty advanced!

What is a COM object?

COM (Component Object Model) is described by Microsoft as "platform-independent, distributed, object-oriented system for creating binary software components that can interact". The purpose of this technology is to provide an interface to allow developers to control and manipulate objects of other applications. We already spoke about this technology in the IcoScript case. Each COM object is defined by a unique ID called CLSID. For example the CLSID to create an instance of Internet Explorer is {0002DF01-0000-0000-C000-000000000046}.

COM object hijacking analysis

During the installation phase, the malware drops two files into the directory:
%APPDATA%\Roaming\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}\
The file names are created using the following scheme: api-ms-win-downlevel-[4char-random]-l1-1-0._dl
One file is the 32-bit version of the malware and the second one is the 64-bit version.

The second step: the creation of two registry entries:

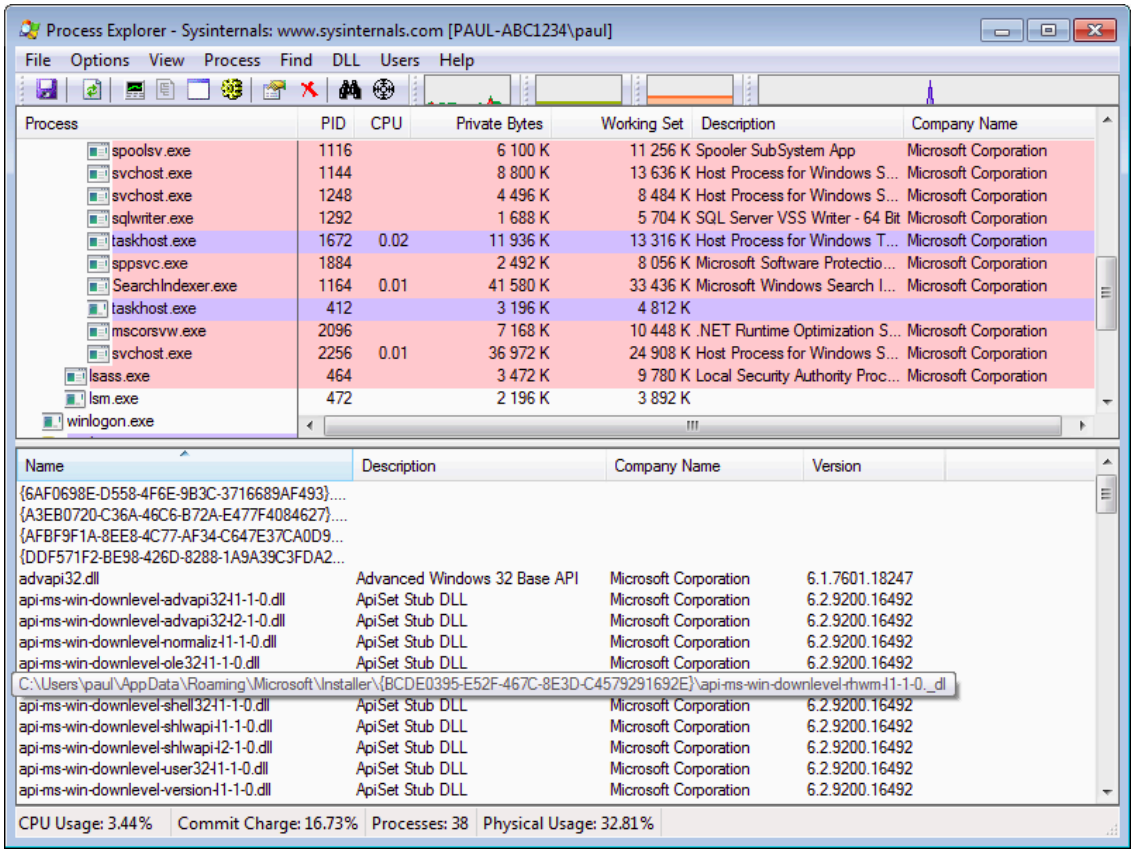
- HKCU\Software\Classes\CLSID\{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}\InprocServer32
- HKCU\Software\Classes\Wow6432Node\CLSID\{BCDE0395-E52F-467C-8E3D-C4579291692E}\InprocServer32

For each entry, the default value is the path to the files that were dropped before. In the following screenshot, the file containing rhwm is the 64-bit version of the malware and the file containing dtjb was created for the 32-bit version, respectively.

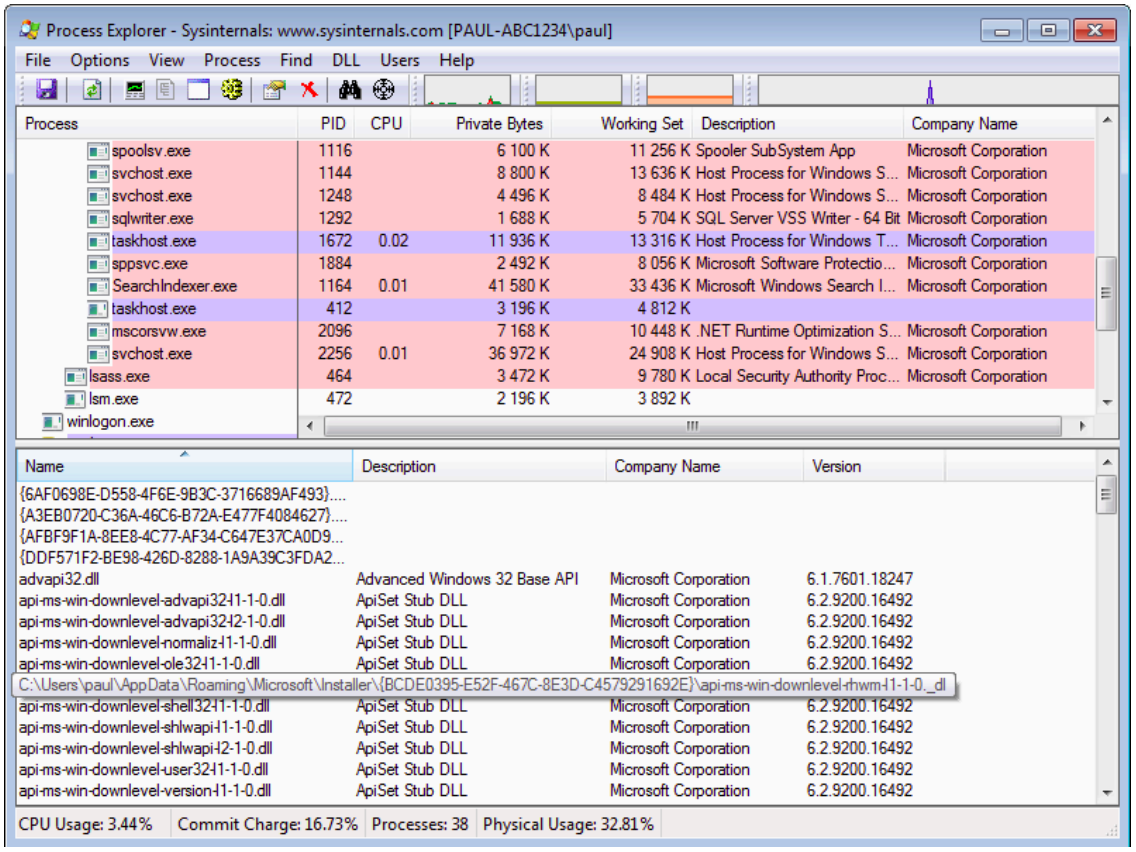
The purpose of the keys is to define a COM object with the CLSIDs {b5f8350b-0548-48b1-a6ee-88bd00b4a5e7} and {BCDE0395-E52F-467C-8E3D-C4579291692E}. If these objects are instanced, the library will be loaded into the respective process. But the CLSIDs are predefined by Microsoft and the newly created owns replace the originals:

- {b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}: the CLSID of the class CAccPropServicesClass.
- {BCDE0395-E52F-467C-8E3D-C4579291692E}: it's the CLSID of the class MMDeviceEnumerator.

These two instances are used by a lot of applications, for example by the browser (by using the CoCreateInstance() function). With Process Explorer, we are able to list the library loaded into a specific process. Here are the loaded libraries designed for a 32-bit process:



The following screenshot shows the loaded libraries in a 64-bit process:



In both of these cases, we can see our dropped library. The processes use the registry key previously created to load the malicious library instead of the original Microsoft library

Conclusion

This new approach of persistence mechanism has several advantages: the attacker does not need to perform DLL injection, which is usually monitored by anti-virus software. Therefore, he has overcome one important security measure, in most of the cases.
As soon as the infection was successful, Microsoft Windows then natively executes the library in the processes of the infected user. Hence, the attacking process is hard to be identified. Using COM hijacking is undoubtedly silent. It is not even detected by Sysinternals' Autoruns.

So, in our case, we have seen this mechanism being used combined with a RAT and this would mean a hassle for any infected and therefore affected user, as the attackers can spy on him pretty secretly for quite some time. But, obviously, this new persistence mechanism is not limited to the use of RATs. Attackers can combine it with any other type of malware, too!

G DATA customers are protected: the latest scan engines detect the analyzed samples (see below). Furthermore, our behavior blocker technology identifies this threat and blocks it.

IOC

Registry

- HKCU\Software\Classes\CLSID\{b5f8350b-0548-48b1-a6ee-88bd00b4a5e7}\
- HKCU\Software\Classes\Wow6432Node\CLSID\{BCDE0395-E52F-467C-8E3D-C4579291692E }

Files

- %APPDATA%\Roaming\Microsoft\Installer\{BCDE0395-E52F-467C-8E3D-C4579291692E}\
  The file names are created using the following scheme: api-ms-win-downlevel-[4char-random]-l1-1-0._dl

MD5

- 482a70b7f29361665c80089fbf49b41f
  (Trojan.Generic.11683196; Win32.Trojan.COMpfun.B)
- 88fc61bcd28a9a0dc167bc78ba969fce
  (Trojan.Generic.11671459; Win32.Trojan.COMpfun.A)
- 11f814e7fb9616f46c6e4b72e1cf39f6
  (Gen:Trojan.Heur2.LP.dq8@aKnXWtic; Win32.Trojan.COMpfun.C)
- 671d230ae8874bb89db7099d1c8945e0
  (Win64.Trojan.COMpfun.C)

Side note:

You are maybe wondering about the malware signature name containing “fun”. During our analysis, the [4char-random] value of the malware name was “pfun” and… yeah, we thought that this was ‘fun’ny, indeed ;-)

Share Article
