How the Silence Downloader Has Evolved Over Time

norfolkinfosec.com/how-the-silence-downloader-has-evolved-over-time/

norfolk February 11, 2019

In a previous post this blog briefly compared two versions of the Silence group's proxy malware, a post-intrusion tool used to relay network traffic between a C2 endpoint and a non-internet facing device. This post examines three versions of the group's downloader and documents how it has changed over the last eighteen months. While some characteristics have persisted, several notable functions have been removed, added, or modified in newer versions of this tool.

Tracking such changes helps analysts determine whether or not a newly discovered sample (on the network or in an online repository) is truly new; in the event that the sample is older and forensic data is missing, it can help approximate when the sample might have been deployed.

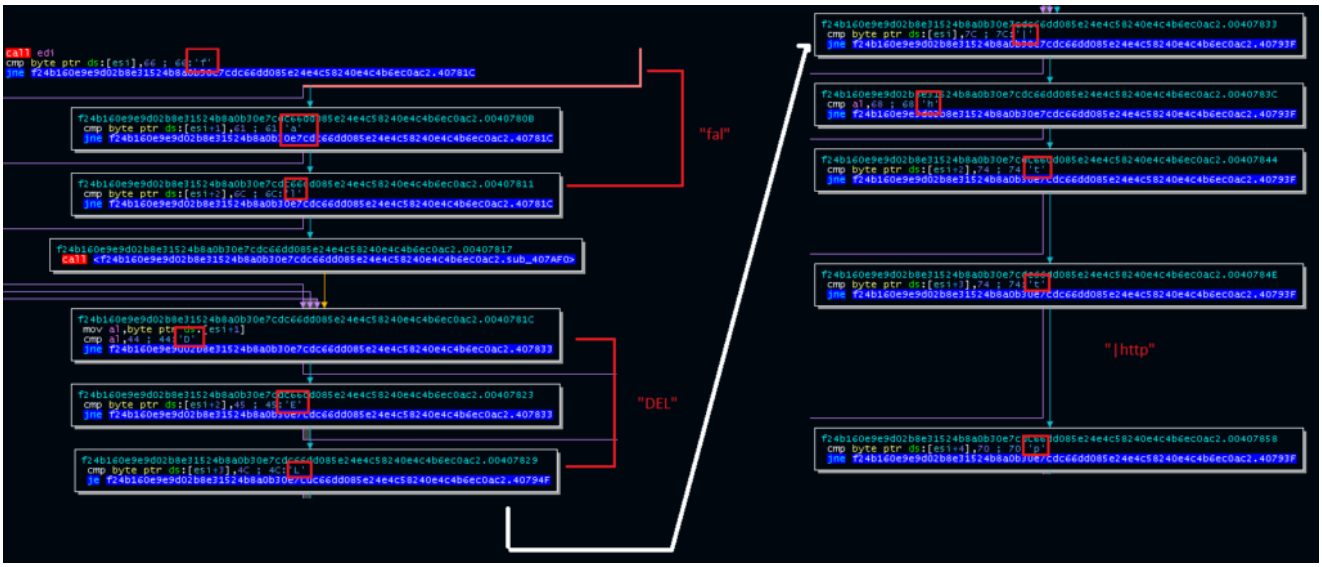
October 2017

MD5: 404D69C8B74D375522B9AFE90072A1F4
SHA1: 197d8bc245ba8b67ebf9a108d6707011fe8158f9
SHA256: f24b160e9e9d02b8e31524b8a0b30e7cdc66dd085e24e4c58240e4c4b6ec0ac2

This Silence downloader was first publicly described at a high level in a Kaspersky Securelist post in October 2017. The downloader calls out to a C2, and the response allows it to:

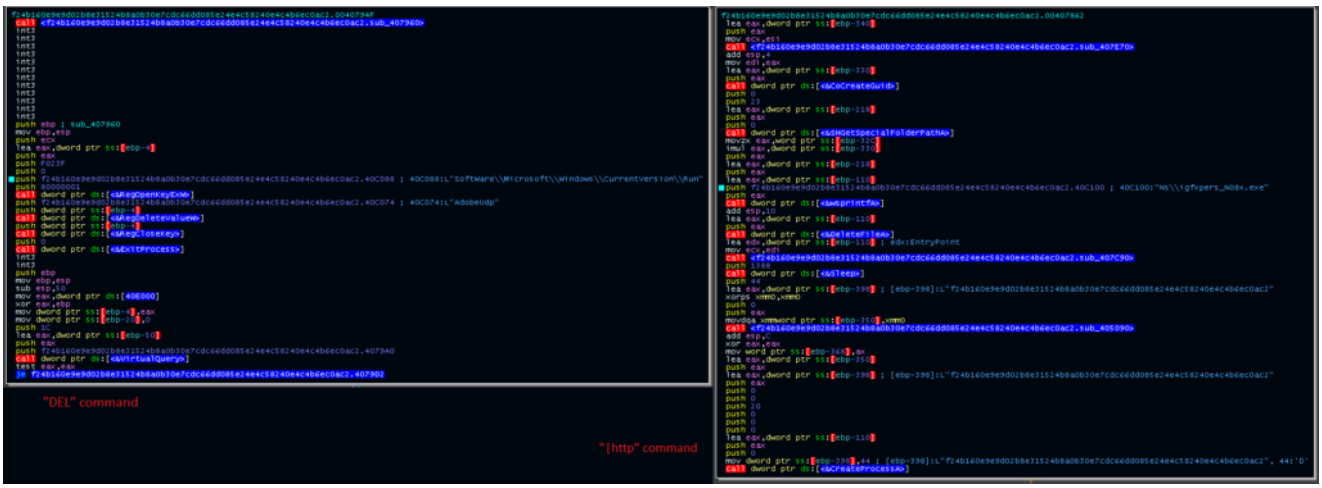
– Create an auto-start persistence entry in the registry (HKCU CurrentVersion\Run) for a copy of itself ("fal")
– Obtain an additional payload, save this payload to disk, and execute it ("|http")
– Delete itself ("DEL")

As this malware serves as a simple, early-stage tool, these tasks (and their underlying mechanisms) have gone largely unexamined in the public space; however, there are several distinct characteristics regarding how the earlier versions of the malware accomplish this workflow. The figure below shows the malware's logic flow prior to taking one of the actions above:



October 2017 Silence downloader logic flow

Rather than comparing the bytes (or their corresponding strings) as a whole, the malware performs a byte-by-byte check of the action, jumping over the remainder of the comparisons should a byte not match. If no task is identified, the malware sleeps and attempts to retrieve a task from the C2 a second time. The figure below shows the functions called following a successful parsing of the “|http” or “DEL” actions.



DEL and |http functions from the October 2017 Silence downloader

November 2018 Sample

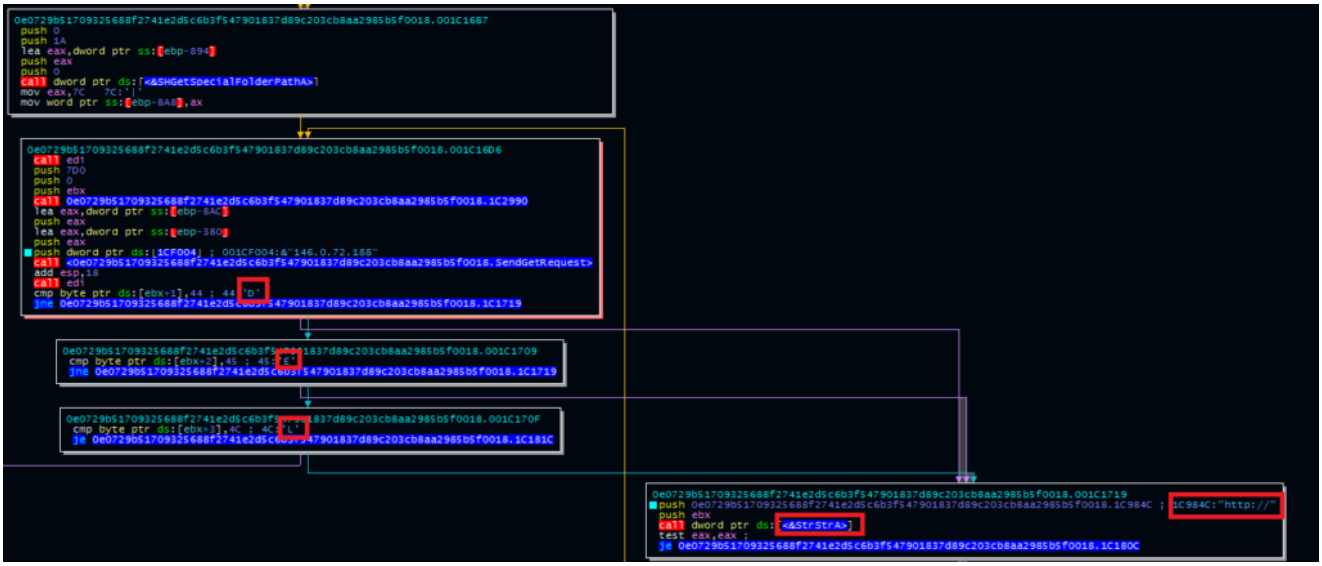
Late last month, Reaqta published research that included details of a late-2018 version of the Silence downloader. At a high level, the downloader includes a key addition: the downloader executes a series of command-line queries to obtain information about the infected device. This information is stored locally in the user’s ProgramData folder in a file named “INFOCONTENT.TXT” and uploaded to the C2 server. Interestingly, a handful of the command-line commands are initially obfuscated, though several others remain in clear text.

Command-line information collection

While this is a key addition on its own, the author(s) of the tool also made two notable changes to the tasking workflow:

– The “fal” action used to create persistence has been removed. The tool now takes this step without prompting.
– The “|http” action still exists; however, it is no longer initiated by a byte-by-byte comparison. Instead, the authors opted to use the StrStrA function to determine if “http” is in the task string.

Curiously, the authors did *not* change the “DEL” task initiation to align with the change to “|http.” It still uses the same single-byte comparison and jump. It’s possible that the authors were either testing the new mechanism first or hadn’t yet had time to change both functions.



November 2018 sample. Left: “DEL” single-letter comparisons remain intact. Right: “|http” comparison replaced with StrStrA call.

Although the mechanism for calling the persistence routines changed between versions, the routines themselves are largely consistent:

Left: 2017 persistence routine. Right: November 2018 persistence routine.

Late 2018/Early 2019 Samples

MD5: e2e1035f382c397d64303e345876a9db
SHA1: c572ba3fcd991fd29919d171b8445dbb5277a51d
SHA256: 4ea01c831c24b70b75bcdf9b33ad9c69e097cbadafd30599555a43a1f412455d
C2: 185.244.131[.]68

Pivoting through VirusTotal using the string “%s\%08x%08x.tmp” from the previous sample leads to a new set of updated downloaders from this threat actor. These more recent samples contain significant changes, including:

– A revised mechanism for establishing the registry-based persistence mechanism
– An alternate persistence mechanism using depending on the detected operating system
– An antivirus check to facilitate this check
– The ability to execute a payload OR register a DLL

The screenshot below depicts the version check alongside the AV check. Notably, the authors implemented an AV check that calls CreateToolhelp32Snapshot, Process32First, and Process32 next for each string, rather than calling each of these up front and then performing the string comparison.

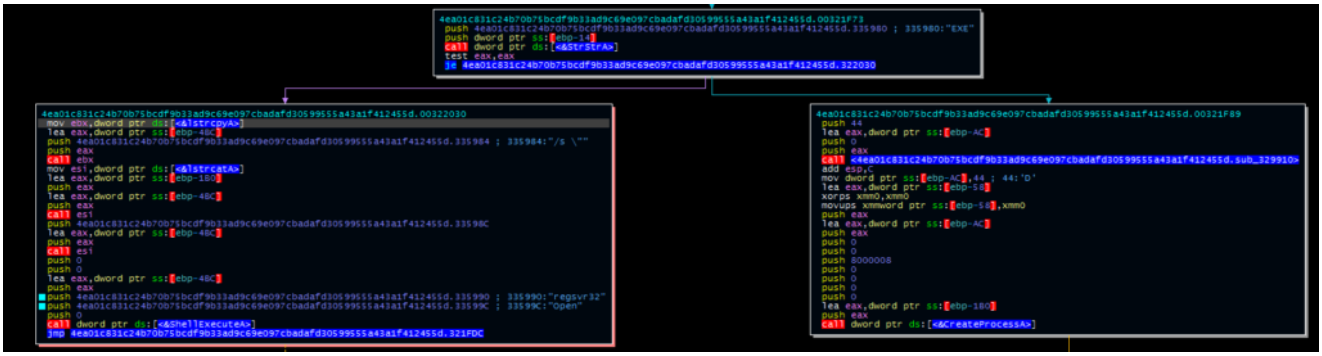
AV and Operating System Checks

Depending on the results of the OS detection and the AV check, the malware can create a registry entry for persistence or create a scheduled task. Unlike the previous versions, the registry entry is not created through API calls; instead, the malware decodes and executes a command line string:

"/C REG ADD "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "Windows System DLL"

Alternatively, in this particular sample, the malware can create a scheduled task named "Avi Capture."

Finally, this version features an updated version of the C2 mechanism. The malware now uses a different set of APIs (Winsock) to contact the C2 and receive a response. The downloader can both register a DLL or launch an executable file in this version. True to the previous two versions, this workflow is initiated by a byte-by-byte check for "MZ" (a PE header) in the file.



Workflow for DLL vs EXE execution in Late 2018/Early 2019 Silence Downloader

Concluding Thoughts

Tracking how a malware family changes over time helps categorize how “new” a newly uploaded or discovered sample really is. While compilation timestamps can be spoofed, a threat actor is unlikely to revert to a previous version of a tool, particularly if it contains errors or lacks required features.

In the Silence downloader, it is apparent that the threat actors have taken an interest in collecting a larger set of initial information (including operating system data). In addition, the threat actors have also taken several measures to evade or bypass AV detection, including basic process checking and string obfuscation.