Trickbot

Alföldi Mátyás

June 24, 2020

Contents

Introduction

File: imgpaper.png
VT: https://www.virustotal.com/gui/file/934c84524389ecfb3b1dfcb28f9697a2b52ea0ebcaa510469f0d2d9086bcc79a/detection

Initial VT report check

Sections:

- .text section seems to have a decently high entropy, so there will most likely be an unpacking process.
- .rsrc section is quite big, with a close to 8 entropy, signaling that it is definitely packed.

Interesting Imports:

- advapi32.dll
  - Various registry related api calls + Get/SetFileSecurity
- kernel32.dll
  - Virtual* functions
  - Resource handling functions (for the unpacking)
  - Heap related functions
  - File operations (like Read/Write file)
  - Library loading/unloading
  - SetUnhandledExceptionFilter + UnhandledExceptionFilter combo
  - Various Get* functions for information gathering
- WINSPOOL.DRV (This in itself is strange)

Static Analysis

0.1 imgpaper.png

Once entering WinMain it immediately gets the current thread with AfxGetThread, the returned value is later on used, to calculate function pointers to call(By adding x to the returned value).
Afterwards it tries to Load a string resource, which doesn't seem to exist.
Installs a hook regarding input into dialogbox,msgbox,etc. for the current thread.
In between there is a calculated func ptr call, and it undo-s the hook and calls UnregisterClassA.
Here will most likely the interesting part start, since after the WinMain there is a bunch of calculated func ptr call, and there might be some anti-debugging code.
The RCDATA in the .rsrc section seems to be the cause for thehigh entropy, most likely this stores some PE file.
Preparing for debugging it:
As one can see, it will create a child process, so setting a bp on CreateProcessInternalW/A, VirtualAlloc, VirtualProtect, WriteProcessMemory, NtWriteVirtualMemory will most likely help us catch the probably unpacked PE file that gets inserted into some system process. (Most likely Process Hollowing will take place.)
Debugging:
It seems to store some system info in the first VirtualAlloc-ed location.
And just as I thought the second VirtualAlloc-ed location has a PE file without the MZ magic number(Although at the first VirtualProtect time, it only has a the .text section ready).
Afterwards the next VirtualAlloc-ed location has another PE in it(check if they are the same). This one is already detected on virustital by a few engines:
https://www.virustotal.com/gui/file/b19fda07723593cd078e14ebec6ef535256a44bd5377c96d044923406374d158/detection
Next it halted on CreateProcessInternalW, where it tries to start wermgr.exe, and obviously it is started in suspended state, but it seems to not use WriteProcessMemory, or NtWriteVirtualMemory to do the copy over.
Also it is wise to attach a debugger, to the suspended process, since the main process exits.