Parent PID Spoofing (Stage 2) Ataware Ransomware - Part 0x3 - Securityinbits

By Ayush Anand

Published: 2019-05-14 · Archived: 2026-04-05 16:49:14 UTC

Ataware Ransomware Stage 2 uses Parent PID Spoofing technique to change its parent PID to lsass.exe and this article is also referred to in Mitre Attack website [4]. You may download the ATAPIConfiguration.exe file from ANY.RUN (MD5: 04a2e6400b22a3a5e5e277eceaf2ce0c)

Overview of ATAPIConfiguration.exe (Stage 2)

Stage 2 downloads the final Ataware Ransomware (ATAPIUpdtr.exe) from CC which can encrypt files. Then, it uses Parent PID Spoofing to change the parent PID to lsass.exe before executing it.

CONTENTS

Static Analysis

- 32bit PE, compiled using GCC MINGW
- Nothing interesting in overlay, no resources
- Compiler timestamp invalid is 1997
- File contain TLS callback but nothing interesting

Strings

Based on the strings berylia[.]net and /index/, we can guess that malware may be downloading something.

```
 wininet.dll
InternetConnectW
berylia.net
HttpOpenRequestW
/index/
GET
InternetQueryOptionW
InternetSetOptionW
```

`HttpSendRequestA`
`TEMP`
`kernel32.dll`
`CreateFileW`
`InternetReadFile`
`WriteFile`
`InternetCloseHandle`
`Advapi32.dll`
`LookupPrivilegeValueW`
`AdjustTokenPrivileges`
`OpenThreadToken`
`ImpersonateSelf`
`SeDebugPrivilege`
`CreateToolhelp32Snapshot`
`Process32FirstW`
`Process32NextW`
`lsass.exe`
`OpenProcess`
`InitializeProcThreadAttributeList`
`UpdateProcThreadAttribute`
`CreateProcessA`

Parent PID Spoofing

Stage 2 mainly uses **InitializeProcThreadAttributeList**, **UpdateProcThreadAttribute** & **CreateProcessA** with *STARTUPINFOEXA* structure API for spoofing. Didier Stevens already blogged about this in 2009 [1], "Normally the parent process of a new process is the process that created the new process (via **CreateProcess**). But when using *STARTUPINFOEX* with the right **LPPROC_THREAD_ATTRIBUTE_LIST** to create a process, you can arbitrarily specify the parent process, provided you have the debug rights." Before spoofing, this Stage 2 enables **SeDebugPrivilege** of current thread.

UpdateProcThreadAttribute function[2] is called with *PROC_THREAD_ATTRIBUTE_PARENT_PROCESS* (0x00020000) attribute with the handle of lsass.exe. At last, **CreateProcessA** is called with *STARTUPINFOEXA* Structure which contain new *StartupInfoEx.lpAttributeList* and creation flag 0x80010 (*EXTENDED_STARTUPINFO_PRESENT* | *CREATE_NEW_CONSOLE*) for creating new process with different Parent PID.

Analysis steps in Ghidra

1. Navigate to entry function, then to WinMain address @ *0x4013dd* as shown below

2. We will concentrate on **download_spoof_parent_process_exe** (*0x40208b*) in main function as shown below

3. Before any rename/comment @ *0x401cb7*

4. This function contains two main functions FUN_00401b91() & FUN_00401579().
5. Let's focus on FUN_00401b91 (**adjust_priv_current_thread_sedebug**), this function enables **SeDebugPrivilege** of current thread.

grade

Tip: **LookupPrivilegeValueW** & **AdjustTokenPrivileges** API are very common in malware when they want to enable **SeDebugPrivilege** privilege. For details, please check this msdn [3].

6. FUN_00401579()/**download_save_ATAPIUpdtr_exe** function download the file from CC **hxxps://berylia[.]net/index/** and save it to $temp directory as ATAPIUpdtr.exe.

7. Parent PID Spoofing is shown below in the final code **download_spoof_parent_process_exe**.

Dynamic Analysis using Sysmon

File Create event Sysmon

Here you can see Process Create with spoofing in action with Parent Image lsass.exe.

Conclusion

- Analysed Parent PID Spoofing and saw this in action using Sysmon
- Malware uses this technique to evade detection which is based on parent-child process
- We understood how malware author can enable SeDebugPrivilege

Thanks for reading. Feel free to connect with me on or LinkedIn for any suggestions or comments.

For more updates and exclusive content, subscribe to our newsletter. Stay sharp. Keep defending.😊

Join 150+ subscribers who get 0x1 actionable security bit every week.
