By: AlSch092 For: MITRE ATT&CK

Technique Name: Change Module Names in Running Processes

Tactic: Defense Evasion

Platform: Windows

Required Permissions: User

Sub-techniques: This is a technique of TA0005.

Data Sources: Windows API, Process Environment Block

Description:

The names of loaded modules in a process can be modified at runtime to avoid detection mechanisms. This is done by determining the address of a module's string name and then writing another value over it. Any process can perform this technique on itself or other processes as long as the memory where module names are located is writable.

In the context of a running process, calls to the Windows API `GetModuleHandle` will return NULL if one queries a module name which has been changed previously by this technique, which potentially increases the evasion abilities of a module. Program behavior may also be altered on the basis that `GetModuleHandle` returns NULL. Loaded modules names can also be changed to the same or duplicate values, making it harder to determine which module is the original.

This technique can also be used to hijack or intercept program execution. If a process queries the address of a module which has had it's name replaced with a malicious one, the malicious module can potentially export a function with the same name and parameters as one that is looked up and called by the victim process.

Example (Availability): A malicious process could change the module name of "KERNEL32.dll" inside a remote process to "USER32.dll", resulting in two versions of "USER32.dll" being loaded, and calls to `GetModuleHandle("KERNEL32.dll")` would then return NULL. Because the module name can no longer be found using API, program behavior may change or create availability issues.

Example (Deception): A malicious process named "Quasar.exe" renames its module in memory to "Skype.exe", thus when programs query the list of modules for any blacklisted names they are presented with one that looks like a trusted Windows program.

Example (Persistence): A malicious actor side-loads a DLL (T1574.002) and then changes the DLL's name to an empty string, increasing the chances that no further forensic action will be taken.

Detection:

Read the entire path including the file name when querying loaded modules, and check for the existence

of the module's file name at the path's location.

If two or more of the same module name is found loaded in a running process, then it means at least one of those modules had their names modified.

Mitigation:

Ensure that memory is non-writable for locations on the heap where module string names reside at.

Save the names of all loaded modules and their memory addresses at program startup, such that if any are later modified it can be clearly determined.

Adversary Use: No examples could be found as this is a newly discovered technique. Further data must be collected to determine if any past malware samples have used this technique.

Additional References: Here is a reference from the researcher who discovered this technique, which includes a description and code examples: (https://github.com/alsch092/ChangeModuleName)

A published and peer-reviewed reference to this technique can also be found at: (https://unprotect.it/technique/change-module-name-at-runtime/)