./ persistence-info.github.io

View on GitHub

Windows Platform Binary Table

Location:

UEFI

Classification:

Criteria	Value
Permissions	Other[1]
Security context	System
Persistence type	Other
Code type	EXE[2][3]
Launch type	Automatic
Impact	Non-destructive
OS Version	All OS versions
Dependencies	OS only
Toolset	Own toolkit required

Description:

Hardware-based persistence.

1. During the OS startup, `smss.exe` calls `NtQuerySystemInformation()` function with a `SystemPlatformBinaryInformation` (0x85) as a parameter.
2. NtQuerySystemInformation() scans UEFI tables stored within hardware memory looking for a piece of data with properly constructed headers.
3. If the correct pattern (“`WPBT`”, length, revision and a checksum) is found, the structure is passed to the `smss.exe`.
4. `smss.exe` stores the piece of UEFI memory within a file called `%systemroot%\system32\wpbbin.exe`.
5. `smss.exe` takes execution parameters (command line) from the same UEFI block.
6. The `wpbbin.exe` is checked for integrity with `IMAGE_DLLCHARACTERISTICS_FORCE_INTEGRITY`.
7. The `wpbbin.exe` is executed.

The functionality may be disabled with the `DisableWpbtExecution` registry value set to `1` in `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager` (tip by @Harvesterify)

The functionality is not a typical persistence, as it does not rely only on configuration stored within Windows. As written above, the exploitation requires both: writing into UEFI tables AND digital signature meeting `IMAGE_DLLCHARACTERISTICS_FORCE_INTEGRITY` requirements.

References:

>> http://download.microsoft.com/download/8/a/2/8a2fb72d-9b96-4e2d-a559-4a27cf905a80/windows-platform-binary-table.docx
>> https://grzegorztworek.medium.com/using-uefi-to-inject-executable-files-into-bitlocker-protected-drives-8ff4ca59c94c
>> https://github.com/tandasat/WPBT-Builder

Credits:

@Harvesterify

See also:

Remarks:

1. File content is stored within UEFI tables. ↩
2. `wpbbin.exe` is created on disk during boot process ↩
3. The code must rely on `ntdll.dll`, without any Win32 API calls. ↩