

Open Core Windows Blue Screen

Beitrag von „griven“ vom 4. Juli 2020, 23:32

Grundsätzlich übernimmt OpenCore alle Änderungen die an den ACPI Tabellen vorgenommen werden für alle Betriebssysteme die über OpenCore gestartet werden und damit ist eigentlich schon die Erklärung für den BlueScreen von Windows gefunden. Viele Tutorials, die so im Netz rumschwirren, sind leider oftmals ein gutes Beispiel dafür wie man ACPI Patches **nicht** macht. Schaut man mal in Deinen EFI Ordner finden sich dort zwei ACPI Dateien einmal eine DSDT.aml und eine SSDT-EC0.aml bei der DSDT kann ich natürlich nicht sagen was daran verändert wurde hier kenne ich das Original nicht daher lassen wir die mal außen vor (vermutlich wird sie eh nicht gebraucht) aber anhand der SSDT-EC0.aml kann man schön aufzeigen wie man so etwas so gestalten kann das es sich nur auf macOS auswirkt und alle anderen Betriebssysteme ignoriert, was es im vorliegenden Fall auch tut. Die von Dir eingesetzte Version sieht wie folgt aus:

Code

```
1. DefinitionBlock ("", "SSDT", 2, "ACDT", "EC0", 0x00001000)
2. {
3.   External (_SB_.PCI0.SBRG.EC0_, DeviceObj)
4.
5.   Scope (\_SB.PCI0.SBRG.EC0)
6.   {
7.     Method (_STA, 0, NotSerialized) // _STA: Status
8.     {
9.       If (_OSI ("Darwin"))
10.    {
11.      Return (Zero)
12.    }
13.   Else
14.   {
15.     Return (0x0F)
16.   }
17. }
18. }
19. }
20.
```

Alles anzeigen

auffällig dabei ist der folgende Block:

Code

1. If (_OSI ("Darwin"))
2. {
3. Return (Zero)
4. }
5. Else
6. {
7. Return (0x0F)
8. }

Die IF Bedingung unterscheidet in dem Fall ob das Betriebssystem macOS (Darwin) ist oder nicht und im Falle von macOS gibt die Methode den Wert NULL zurück andernfalls den Wert 0x0F wobei der "andernfalls" Wert dem entspricht was auch zurückgegeben würde wenn es die SSDT nicht geben würde. Das Beispiel zeigt wie man ACPI Patches so anfertigt das sie sich nur auf ein bestimmtes Betriebssystem auswirken. Wie wir sehen entspricht die SSDT-EC0.aml den notwendigen Standards und kann daher nicht der Grund für den ACPI Fehler sein den Windows anmeckert. Das Problem ist also irgendwo in der DSDT zu suchen und hier ist es leider nahezu unmöglich einen Diagnose zu stellen ohne die originale DSDT zu kennen. Wie auch immer ich würde an Deiner Stelle die DSDT erstmal aus dem Rennen nehmen und gucken ob Windows damit zufrieden ist und falls das der Fall ist (ich gehe davon aus) dann würde ich im nächsten Step gucken ob auch macOS ohne die DSDT auskommt und wenn auch das der Fall sein sollte (ehrlich gesagt gehe ich auch davon aus) dann kannst Du Dich langsam ran tasten und ausprobieren was unter macOS mit der DSDT funktioniert hat und ohne nicht mehr und gezielt diese Dinge adressieren. Heute braucht man eigentlich so gut wie gar nicht mehr an der DSDT rum schrauben weil vieles von dem was man da mal geregelt hat sich viel eleganter anderweitig regeln lässt (Stichwort IOREG bzw. Lilu.kext und Co.)...