

Scan Policy berechnen

Inhaltsverzeichnis

- [1 Viele bzw. genauer gesagt - zwei Wege führen jetzt nach Rom - äh zum Ziel.](#)
 - [1.1 Weg #1 - addieren der HEX-Werte.](#)
 - [1.2 Weg #2 - setzen der benötigten Bits](#)
 - [1.3 Wichtiger Hinweis:](#)

Hi zusammen,

da das Wort "[Scanpolicy](#)" hier schon fast 100 Ergebnisse in der Suche bringt und ich es auch selber nicht kapiert habe, versuche ich das hier für die Nachwelt mal transparent darzustellen.

Ggf. ist es auch für diejenigen nützlich, die nicht ausreichende Englischkenntnisse haben.

Wie immer bei OC gilt: Wenn man es einmal verstanden hat, ist es irgendwie gar nicht mehr kompliziert. 🧐

Trotzdem kann es sein, dass ich unten etwas falsch wiedergebe, da ich es ja gerade erst erklärt bekommen habe.

Sollte also jemand Fehler oder auch ungünstige Umschreibungen finden, bitte bei mir melden, damit ich es korrigieren kann.

Großen Dank an dieser Stelle geht an @kuckuck für die super Erklärungen und die Geduld, meine doofen

Fragen zu beantworten



Los gehts ... aber ein wichtiger Hinweis noch: Wer kein Dualboot-System hat - und somit im Notfall immer in der Lage ist, z.B. via Windows die [Scanpolicy](#) wieder zu ändern, sollte die folgenden OC-Funktionen nur per USB-Stick testen.

Andernfalls besteht die Möglichkeit, dass man sich selbst ausschließt - das wäre - sagen wir mal - suboptimal 🤔

Um zu sehen, was überhaupt im "best-case" gescannt wird, einfach einmal die [Scan Policy](#) = 0 setzen. So wird einfach alles gescannt und angezeigt, was automatisch gefunden wurde.

Als Basis dient immer die Dokumentation von OpenCore - besonders der Punkt 8.5 (Security Properties) - Unterpunkt 7.

Hier wurden die Informationen mal in die u.a. Tabelle übernommen (**Stand OC-Dokumentation 0.5.8**)

Da ich nicht immer so schnell sein werde, wie es OC Updates gibt, bitte immer mit der aktuellen Dokumentation vergleichen.

Die jeweils aktuelle Configuration.pdf von OpenCore findest du >>[HIER](#)<<

HEX-Wert	Bit	Funktion / Hinweis
0x00000001	bit 0	Dieser Bit muss in Abhängigkeit zu den gewünschten Einstellungen (Filesystem) gewählt werden. Sprich möchtet ihr den Scan auf die gewählten Filesystem einschränken, MUSS dieses Bit gesetzt sein!
0x00000002	bit 1	Dieser Bit muss in Abhängigkeit zu den gewünschten Einstellungen (Device) gewählt werden. Sprich möchtet ihr den Scan auf die gewählten Geräteklasse/Device einschränken, MUSS dieses Bit gesetzt sein!
0x00000100	bit 8	Dieser Bit muss gesetzt sein, wenn man das Scannen nach APFS wünscht - z.B. für macOS oder CarbonCopy
0x00000200	bit 9	Dieser Bit muss gesetzt sein, wenn man das Scannen nach HFS wünscht - z.B. für ältere macOS Versionen
0x00000400	bit 10	Dieser Bit muss gesetzt sein, wenn man das Scannen nach EFI wünscht - z.B.
0x00000800	bit 11	Dieser Bit muss gesetzt sein, wenn man das Scannen nach NTFS wünscht - z.B.
0x00001000	bit 12	Dieser Bit muss gesetzt sein, wenn man das Scannen nach EXT wünscht - z.B. für LINUX Systeme.
0x00010000	bit 16	Dieser Bit muss gesetzt sein, wenn man das Scannen auf SATA Devices wünscht.
0x00020000	bit 17	Dieser Bit muss gesetzt sein, wenn man das Scannen auf SAS oder Mac NVMe Devices wünscht.
0x00040000	bit 18	Dieser Bit muss gesetzt sein, wenn man das Scannen auf SCSI Devices wünscht.
0x00080000	bit 19	Dieser Bit muss gesetzt sein, wenn man das Scannen auf NVMe Devices wünscht.
0x00100000	bit 20	Dieser Bit muss gesetzt sein, wenn man das Scannen auf CD/DVD Devices wünscht.
0x00200000	bit 21	Dieser Bit muss gesetzt sein, wenn man das Scannen auf USB Devices wünscht.
0x00300000	bit 22	Dieser Bit muss gesetzt sein, wenn man das Scannen auf FIREWIRE Devices wünscht.
0x00400000	bit 23	Dieser Bit muss gesetzt sein, wenn man das Scannen auf SDCARD Devices wünscht.

1 Viele bzw. genauer gesagt - zwei Wege führen jetzt nach Rom - äh zum Ziel.

1) Man addiert die HEX-Werte

2) Man aktiviert einfach die jeweiligen Bits

Für beide Wege brauchen wir jetzt den nativen macOS Rechner.

Sobald der gestartet ist, müsst ihr diesen auf den "Programmierer-Modus" wechseln. Das macht ihr entweder mit `CMD+3` oder über das Menü (Darstellung).

Danach drückt ihr noch auf den Button `Binärwerte anzeigen` und auf den Button mit der `16` - für Hexadezimal.

Um es jetzt transparenter zu machen, setzen wir uns ein Ziel:

Wir möchten nur nach **NVMe Devices** mit dem **Filesystem APFS** scannen

Genauer gesagt, ist die Definition der Scanpolicy dann wie folgt - das ist jetzt etwas komplizierter, aber nur so versteht ihr, wie das Gesamtkonstrukt funktioniert.

Erlaubt das Scannen auf NVMe Devices im APFS Filesystem und verhindert das Scannen jeglicher Geräte mit HFS, EFI, NTFS oder EXT und verhindert das Scannen auf den Devices SATA, SAS, SCSI, CD/DVD, USB, FIREWIRE, SDCARD nach APFS Filesystemen

1.1 Weg #1 - addieren der HEX-Werte.

Wir notieren **100** (APFS) und **80000** (NVME).

Und weil wir damit sowohl nach einem Filesystem als auch nach einem Device scannen wollen, müssen wir auch die **1** (Filesystem) als auch die **2** (Device) notieren.

Jetzt addieren wir alle Werte mit dem Rechner (nicht wundern, wenn sich nach dem Drücken auf `+` (Addieren) sich die Werte ändern - das ist korrekt).

Also $1 + 2 + 100 + 80000 = \underline{\underline{0x80103}}$

Jetzt drückt ihr den Button `10` (für Dezimal) und erhaltet so das Ergebnis `524547` für eure `config.plist`

1.2 Weg #2 - setzen der benötigten Bits

Wir notieren uns die Bits **0** (Filesystem); **1** (Device); **8** (APFS) und **19** (NVMe)

Und genau diese klickt ihr jetzt direkt im Rechner an und bekommt - Überraschung - ebenfalls **0x80103** heraus.

Umgewandelt in Dezimal ergibt das somit auch wieder `524547`

Achtung: Wenn ihr keine Bits seht, habt ihr vergessen den Button `Binärwerte anzeigen` zu drücken

Wenn ihr die Maus auf den Bits lasst, wird euch die exakte Bitzahl angezeigt .. denkt dran - es gibt ein Bit 0



1.3

Wichtiger Hinweis:

Die meisten Bootfiles, wie bootx64.efi oder bootmgfw.efi befinden sich meist auf der Fat32 formatierten EFI Partition.

In diesen Fälle muss man also zwingend auch das Bit 10 respektive den Wert 400 dazunehmen/addieren, sonst sperrt man sich selber aus dem System aus.

Ich hoffe die Anleitung hilft dem ein oder anderen weiter.

Hier noch mal ein Screenshot:

Bildschirmfoto-2020-04-27-um-22-58-50.png

Image not found or type unknown

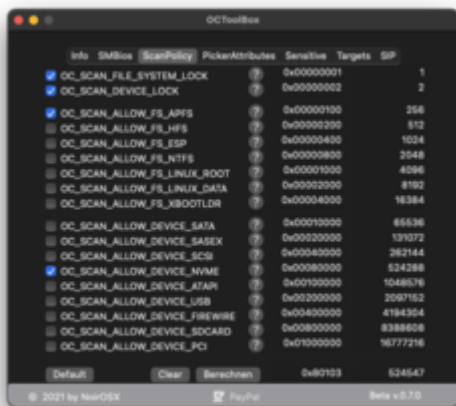
Jetzt, wo ihr das Grundprinzip verstanden habt, versteckt sich unter dem Spoiler die Komplettlösung.



Einfach die Dezimalwerte addieren, als [Scan Policy](#) setzen, fertig.

Um beim obigen Beispiel zu bleiben: $1+2+256+524288= 524547$ 😊

Spoiler anzeigen



[OCToolBox](#)