

Erledigt

[GUIDE] OpenCore Update mit Kext Updater und OCConfigCompare + Config Tweaks

Beitrag von „5T33Z0“ vom 14. September 2020, 20:00

FORTSETZUNG:

4.3 LÖSCHEN veralteter/nicht mehr verwendeter Einträge aus der config.plist

Ich habe im ersten Teil des Tutorials vergessen, dass man am Ende noch die Einträge aus der Config entfernen muss, die im Zuge der Entwicklung gelöscht oder umbenannt wurden. Allerdings sind mir die Zeichen ausgegangen, deswegen mache ich hier weiter.

Wir werfen jetzt einen Blick in den unteren Teil der Liste von OCConfigCompare in der es heißt "Missing from Sample.plist":

hackintosh-forum.de/attachment/151799/

Bei den anderen Einträgen nehmen wir das changelog zu Hilfe. Wir kopieren den Namen der einzelnen Keys und suchen sie im changelog (CMD+F für "Find"):

hackintosh-forum.de/attachment/151800/Wir stellen fest: "ProtectCSMRegion" wurde durch "ProtectMemoryRegions" ersetzt. Jetzt müssen wir prüfen, ob "ProtectMemoryRegions" dieselben Parameter verwendet, wie "ProtectCSMResion". In diesem Fall nicht:

hackintosh-forum.de/attachment/151801/

Also löschen wir in diesem Beispiel "ProtectCSMRegion" und setzen "ProtectMemoryRegion" auf "YES":

hackintosh-forum.de/attachment/151802/

Und so hangeln wir uns durch die restlichen Einträge. Einträge unter "Device Properties" können wir ignorieren. Also zusammenfassend:

- Begriff/Key aus OCConfigCompare kopieren
- Begriff im Changelog suchen
- herausfinden, was mit dem Feature geschehen ist (Moves/Removed? Replaced? Renamed?)

- falls "Removed", schauen, was stattdessen verwendet wird. mit altem Eintrag vergleichen, anpassen und den alten Eintrag aus config.plist löschen
- falls "Replaced/Renamed" prüfen, ob Einstellungen des neuen Features (YES/NO) noch zum ursprünglichen Eintrag passen und ggf. anpassen.

Danach speichern und config erneut überprüfen. Wenn es am Ende so aussieht, sind wir fertig. Einträge zu *DeviceProperties* kann man ignorieren. Ebenso *Type Differences* **wenn** der entsprechende Parameter mehrere Daten-Typen zulässt (bspw. sowohl Data als auch String):

hackintosh-forum.de/attachment/166394/

4.4 OpenCore Config Tweaks

Sicherheitseinstellungen: Den Single User Modus sollte man deaktivieren, da man damit den Passwortschutz des Admin Accounts umgehen kann. Dazu folgenden Eintrag setzen:

Misc > Security > DisableSingleUser = Yes

Bootprobleme beheben: Falls das System nicht startet und direkt beim Bootlogo ohne Fortschrittsbalken hängen bleibt, sollte man folgende Einstellungen ändern:

Misc > Security > Vault = Optional Deaktiviert File Vault. Kann Systemstart verhindern, falls es auf "Secure", steht aber nicht korrekt konfiguriert ist.

Misc > Security > SecureBootModel = Disabled. Ich hatte bisher *immer* Probleme damit, dass Rechner nicht booten, wenn dieses Feature auf "Default" steht. Man sieht dann nur das Apple Logo, aber es passiert nichts weiter. Deswegen deaktivieren, falls es Probleme beim Booten gibt.

System Integrity Protection (SIP) deaktivieren

Unter **NVRAM > Add > 7C436110-AB2A-4BBB-A880-FE41995C9F82** folgenden Wert für **csr-active-config** von **00000000** ändern auf:

FF030000 (für High Sierra)

FF070000 (für Mojave/Catalina)

67080000 (für Big Sur*)

***HINWEIS:** Der im Dortania Install Guide genannte Wert FF0F0000 funktioniert erwiesenermaßen nicht richtig unter Big Sur – man bekommt keine Updates mehr angeboten.

Damit man nach der Änderung des CSR Werts und Neustart nicht jedes Mal einen NVRAM Reset durchführen muss, fügen wir unter **NVRAM > Delete > 7C436110-AB2A-4BBB-A880-FE41995C9F82** **csr-active-config** als Parameter hinzu.

Dieser Wird dann bei jedem Neustart aus dem NVRAM gelöscht und durch den unter "Add" eingetragenen Wert ersetzt. Sehr nützlich wenn man mehrere macOS Versionen installiert hat, die unterschiedliche CSR-Werte verwenden.

Das sieht dann so aus:hackintosh-forum.de/attachment/166821/Um zu testen, ob die Einstellungen nach Neustart übernommen wurden, `csrutil status` nach Neustart ins Terminal eingeben. Das Ergebnis sollte in etwa so aussehen:

Code

1. Configuration:
2. Apple Internal: enabled
3. Kext Signing: disabled
4. Filesystem Protections: disabled
5. Debugging Restrictions: disabled
6. DTrace Restrictions: disabled
7. NVRAM Protections: disabled
8. BaseSystem Verification: disabled

Bootverhalten von OpenCore anpassen

Folgende Einstellungen in der Config ändern, um das Bootverhalten von OpenCore zu verändern. Es gibt natürlich noch mehr Optionen, aber dies sind die gängigsten Varianten.

Startlaufwerk ("Default") im BootPicker ändern:

- **AllowSetDefault** = Yes
- **ShowPicker** = Yes
- Im Boot-Picker: Laufwerk/Partition auswählen
- [STRG] halten und [ENTER] drücken

Apple Hotkey Funktionen aktivieren:

- **PollAppleHotKeys** = Yes

Aktiviert die vom Mac bekannten Tastenkombinationen, um Bootmodi wie Verbose Mode, Safe Mode, Single User Mode, etc. zu nutzen. Details siehe Docs > Configuration.pdf, die dem OpenCore Package beiliegt.

Boot-Option 1: Auswahl des Betriebssystems ohne GUI (Standard)

- **PickerMode** = Builtin
- **ShowPicker** = Yes

Boot-Option 2: Auswahl des Betriebssystems mit GUI (benötigt OpenCanoy und Resource Files)

- **PickerMode** = External
- **ShowPicker** = Yes

Boot-Option 3: immer vom als "Default" definierten Volume starten

- **PickerMode** = Default
- **ShowPicker** = No

Boot-Option 4: automatisch macOS starten, ohne Betriebssystem-Auswahl

- **PickerMode** = BootApple
- **ShowPicker** = No

Verhindert, dass bei deaktiviertem Picker versehentlich Windows oder ein anderes OS gestartet wird, wenn man zuvor vergessen hat, die macOS Disk/Partition als Startvolume ("Default") zu deklarieren. Passiert öfter, als man denkt.

Boot beschleunigen

- **ConnectDrivers** = No

Falls es nach dem Einschalten des Rechners sehr lange dauert (8 Sekunden oder länger), bis

der BootPicker erscheint, kann man mit dieser Option die Wartezeit verkürzen – insbesondere bei Lenovo Notebooks. Allerdings muss man dann auf den Boot-Chime verzichten, da der Audio-Treiber AudioDxe.efi in diesem Fall nicht gestartet wird.

Probleme mit NVRAM Reset beheben

Es gibt BIOS-Varianten, bei denen der integrierte NVRAM Reset von OpenCore Probleme verursachen kann. Symptome: man kommt nicht mehr ins BIOS oder bestimmte Parameter im NVRAM (wie boot-args) werden nicht geändert/angewendet, etc. Daher liegt dem OpenCore Paket unter Tools CleanNvram.efi bei, das soll bei solchen Problemen besser funktioniert. Falls Probleme im Zusammenhang VRAM Reset auftreten, folgendes tun.

- Misc > Security > AllowNvramReset = no (deaktiviert internes NVRAM Reset Tool - ansonsten hat man den Eintrag im BootPicker doppelt)
- CleanNvram.efi nach EFI > OC > Tools kopieren
- neuen Snapshot der Config erstellen oder Tool manuell in Config einpflegen.

Wer möchte, kann man den Eintrag im BootPicker ausblenden, sodass er erst beim Drücken von Leertaste erscheint:

- Misc > Boot > HideAuxiliary = yes
- Unter Misc > Tools "ClenaNvram" finden und "Auxiliary" auf "yes" setzen

Ansonsten würde ich nachschauen, ob es vielleicht ein [BIOS Update](#) gibt, das generelle Probleme behebt.

Falsche OpenCore Versionsinfo korrigieren

Es kann vorkommen, dass die im NVRAM hinterlegte OpemCore Versionsinfo nicht automatisch aktualisiert wird und deswegen in KextUpdater und Hackintool falsch angezeigt wird. Zwar wurde dieses Problem in OC 0.6.7 dadurch behoben, dass die Info gar nicht erst in den NVRAM geschrieben wird, aber sie wird dennoch so lange falsch angezeigt, bis man sie rauslöscht.

Dazu unter **NVRAM > Delete > 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102** ein neues Kindelement erzeugen und als Wert **opencore-version** eintragen und config speichern. Nach Neustart sollte dann wieder die richtige OC Version angezeigt werden.

5. Schritt: EFI Ordner auf USB Bootstick kopieren und Boot testen

- macOS USB Stick Installationsstick oder FAT32 formatierten USB Stick anschließen
- EFI Partion des USB Installationssticks mounten (z.B. mit [MountEFI](#)). Entfällt bei FAT32 formatierten USB Stick!
- EFI Ordner auf EFI-Partition kopieren (die anderen Ordner im Root-Verzeichnis ignorieren):

hackintosh-forum.de/attachment/151809/

- neustarten
- BIOS Bootmenü aufrufen (oft F12)
- USB-Stick auswählen
- Festplatte mit macOS auswählen und starten
- Wenn der Bootvorgang klappt: Herzlichen Glückwunsch!
- Letzter Schritt: EFI-Ordner der Festplatte mounten und durch EFI Ordner auf Bootstick ersetzen

Falls der Bootvorgang fehlschlägt: neustarten von der Festplatte und Backup der originalen config.plist mit der aktualisierten config.plist vergleichen und prüfen, ob und welche Einstellungen abweichen. Diese anpassen, config speichern, auf den USB Stick kopieren und den Boot vom Stick erneut testen. Vorgang wiederholen, bis bis alles funktioniert.

VIEL ERFOLG!

Unterschiede zwischen sample.plists einzelner OC Releases, siehe [Post 11](#)