

Lilu & Plugins mit Bootflags und Beispielen

Beitrag von „Noir0SX“ vom 17. Oktober 2018, 16:55

Helferlein Kext Update & Erstellen

Um die (seine) Kexte aktuell zuhalten, kann man das [Tool \(Kext-Updater\)](#) von [@Sascha 77](#) benutzen

Einen Kext selber erstellen, brauchen hier die wenigsten. Meist sind die Versionen unter Releases zu finden und gut.

Interessant wird es für Leute:

- die Probleme haben, solange in der Repo ein Fix vorliegt
- die nicht warten können
- wie ich das mal probierten
- oder selber den Code verbessern (obwohl der kann auch mit Xcode umgehen)
- usw. ...

Beispiel 1

Am Beispiel von [PMheart](#) Lilu-Plugin [CPUFriend](#), kann man schon sehen, das man einen Kext ganz schnell mal über das Terminal bauen kann.

Als erstes braucht man den Code (Dateien) vom jeweiligen Lilu-Plugin, geht also nicht nur mit dem CPUFriend-Kext.

In das heruntergeladene Verzeichniss laden wir den fertigen Debug-Kext von Lilu.

Das Beispiel CPUFriend zum Ausführen liegt dabei. Zu finden unter Tools mit Namen *build.command*. Diese Datei in das Terminal ziehen und ausgeführt und somit werden dann die beiden Varianten des Plugins im gleichen Verzeichnis erstellt.

Klar auch hier braucht man XCode im Hintergrund, sonst wird das nix.

Der Weg bei den anderen Plugins ist gleich, nur das die Datei unter Tools an den entsprechenden Stellen mit dem jeweiligen Namen des Plugins angepasst und das Verzeichnis Tools kopiert werden muss.

Beispiel 2

Noch einfacher geht das ganze mit den in Python geschriebenen Script von [CorpNewt](#).

[Lilu-and-Friends](#) ist eigentlich selbst erklärend und braucht nach den Download nur ausgeführt werden.

Hier findet man nicht nur Lilu & Plugins, sondern auch viele andere Kexte.

```
#####
# Lilu And Friends #
#####

[ 1. ACPIBacklight - uses ACPI methods to control laptop backlight - pre-OS.X
[ 2. ACPIBatteryManager - implements an ACPI based battery manager kext
[ 3. ATIKMPFixup - for AMD64 (AMD64 & AMD64) AMD64 AMD64
[ 4. AirportBiosFixup - for non-native Airport Broadcom Wi-Fi cards
[ 5. AppleALC - dynamic audio patching
[ 6. AthereusEDMMThermostat - for Miller and Athereus LAN
[ 7. AudioServiceFixup - for Intel's HD4000 mobile GPUs
[ 8. BDM720 - for Broadcom's BCM720 NetXtreme and NetXtreme Family LAN
[ 9. BTLEDeviceFixup - for BTLE-Bluetooth-Hosted Features
[ 10. BrcmPatchRAM - applies PatchRAM updates for Broadcom BCM2046 based devices
[ 11. CPUFriend - for dynamic power management data injection
[ 12. DatasCommander - adds in non-functioning audio after sleep
[ 13. DoremiSleepFixup - enables 4K or greater res on certain graphics
[ 14. EnableLime
[ 15. FakePCIID - spoofed IOPIC devices to allow drivers to load
[ 16. FakeSMC - only FakeSMC
[ 17. FakeSMC (Legacy) - only FakeSMC (Legacy)
[ 18. GenericUSBHID - generic, non-Intel USB support
[ 19. Hibernation (FakeSMC + PlugIn) - contains FakeSMC + Hibernation plugins
[ 20. Hibernation (Legacy) - contains Legacy FakeSMC + Hibernation plugins
[ 21. HibernationFixup - saves IOPIC state to IOPIC in NVRAM
[ 22. IntelBacklight - replaces AppleBacklight for Intel graphics - pre-OS.X
[ 23. IntelGraphicsHWFixup - alternative to the msi-stolen patch
[ 24. IntelGraphicsFixup - provides patches for Intel GPUs
[ 25. IntelMacOSFixup - for most Intel LAN
[ 26. Lilu - for arbitrary kext, library, and program patching
[ 27. NVIDIAGraphicsFixup - unlocks night shift on unsupported GPUs
[ 28. NVIDIAGraphicsFixup - provides patches for NVIDIA GPUs
[ 29. RealtekRTL8100 - for RTL8100 LAN
[ 30. RealtekRTL8100 - for RTL8100/RTL8100 LAN
[ 31. Shiki - for protected video playback
[ 32. USBInjectAll - injects all USB ports
[ 33. VendorPatched - P2P driver for desktops/laptops
[ 34. VendorPatched - change IOPIC to your code (not - 1 in Defs.plist)
[ 35. LiluFriend - Lilu helper for those who keep kexts in /L/I and /L/I/R

B. Build Selected
A. Select All
N. Select None
H. Hide/Build Options
P. Profiles
Q. Quit

Please make a selection: [ ]
```