

Erledigt

macOS Mojave Ozmosis BETA – OZ167X-XMAX

Beitrag von „kuckkuck“ vom 21. Juni 2018, 22:53

Achtung Update 10.14.4+: [UPDATE nötig! – KernnextPatcher 10.14.4+](#)

macOS 10.15 Catalina: [Ozmosis mit macOS Catalina 10.15 - OZ167X-XMAX \(extended\)](#)

Hiermit stelle Ich für Tester ein neues Ozmosis als "Beta" zur Verfügung.

Das ganze richtet sich **ausschließlich** an Tester die wissen was sie tun! Ich übernehme **keinerlei Verantwortung für jegliche Beschädigungen oder Probleme**, die durch die Benutzung dieser Dateien auftreten!

Das ganze ist mit [@cecepawon](#) bei dem Versuch Kext-Injection für Mojave unter Ozmosis zu fixen entstanden. Bei mir läuft alles bereits fabelhaft mit Mojave, was jedoch nichts heißen muss.

Es handelt sich hierbei nicht um eine Art neue Ozmosis Generation, sondern lediglich um ein neues Konzept welches es ermöglicht Mojave mit Kext-Injection zu nutzen. Dazu ist verschiedenes nötig:

Hier steht an erster Stelle eine gepatchte Ozmosis.efi. Um sie von älteren Patches abzugrenzen, geht sie unter dem Namen `Ozmosis_167X-XMAX`. Es findet sich bei dem Konzept eine neue Herangehensweise, um Kexts aus der EFI oder dem ROM in den Kernel zu injecten. Im Anhang:

- EFI Version zur Einbindung mit zB [DBounce: Ozmosis.efi](#)
- FFS Version zum Einbau ins ROM: [Ozmosis.ffs](#)

Die bloße Benutzung von Ozmosis reicht für Kext-Injection nicht aus. Der KernnextPatcher von cecepawon übernimmt hier nun zentrale Aufgaben und **muss** (für Kext-

Injection aus EFI und ROM) **mit Ozmosis zusammen** genutzt werden!

In der FFS Version des KernnextPatcher ist alles nötige für Kext-Injection von *Mountain-Lion bis Mojave* eingebaut. Zukünftige Betriebssysteme lassen sich leicht hinzufügen.

Bei Benutzung des KernnextPatcher aus der EFI (zB mit [DBounce](#)) muss eine passende KernnextPatcher.plist eingebunden werden, welche sich ebenfalls im Anhang finden lässt. Wie dies geht, steht [hier](#). Diese KernnextPatcher.plist kann in jedem Fall durch weitere Einträge erweitert werden, wie KextsToPatch, BooterToPatch oder Kext-Blocks. Weitere Infos dazu [hier](#) und [hier](#).

- FFS Version inklusive eingebauter KernnextPatcher.plist: [KernnextPatcher.ffs](#)
- EFI Version + KernnextPatcher.plist für die EFI-Partition: [KernnextPatcher.efi](#) + [KernnextPatcher.plist](#)

Durch die Benutzung von Ozmosis **und** KernnextPatcher (mit passender Plist) funktioniert Kext-Injection unter Mojave (sowie bei älteren OS X Versionen) von der EFI und aus dem ROM.

Wie man ein Ozmosis-ROM erstellt habe ich hier erklärt: [Ozmosis BIOS individuell anpassen + Erklärung des Aufbaus](#)

In ein aktuelles (Haswell und drunter) ROM sollten folgende FFS eingebaut werden:

- [EnhancedFat.ffs.zip](#) ersetzt den originalen Filesystem-Treiber
- [HfsPlus.ffs.zip](#) sorgt für den Support von HFS-Medien
- [ApfsDriverLoader.ffs](#) lädt den APFS Treiber. v1.1.1 von hier: <https://github.com/acidanthera/ApfsSupportPkg>
- [Ozmosis.ffs](#) gepatcht zur Benutzung mit dem KernnextPatcher
- [KernnextPatcher.ffs](#) inklusive passender Plist, Kext-Injection von Mountain-Lion bis Mojave
- [OzmosisDefaults.ffs](#) als Vorlage. Hinterlegt ist ein iMac14,2 mit aktuellen SMBios-Daten
- [OzmosisHFTheme.ffs.zip](#) oder andere Ozmosis GUIs/Bootmenüs von [hier](#)
- [FakeSMCv1787.ffs.zip](#) die neuste Version von hier: <https://bitbucket.org/RehabMan/os-x-fakesmc-kozlek/>
- evtl. [PartitionDXE.zip](#) für die bessere Unterstützung von (Apple) Filesystem Treibern
- evtl. [AcpiPatcher.ffs](#) da alle fehlerhaften ACPI Header gefixt werden (vergleiche [MATS Problematik](#)). Ebenfalls können viele weitere ACPI-Renames vorgenommen oder CPUStates injected werden: <https://github.com/cecekpawon/...eases-Drivers-AcpiPatcher>
Folgende AcpiPatcher.plist ist in die FFS-Datei eingebaut und sollte angepasst werden: [AcpiPatcher.plist.zip](#) (Bitte [extrahieren](#) und [anpassen](#))

Viel Erfolg an alle Tester, ich freue mich über Feedback, Versuche und Diskussionen! Und wie jeder weiß: Totgesagte leben länger 👍 (und [Nachfolger](#) stehen in den Startlöchern...)