

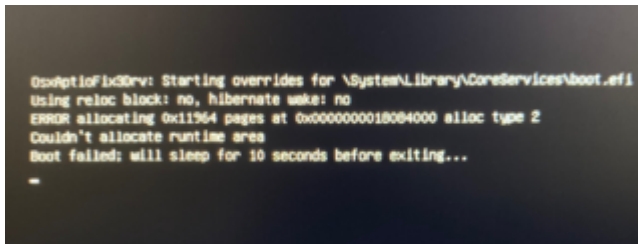
Kein Boot mit aktiver SIP

Beitrag von „macdream“ vom 29. Oktober 2019, 22:33

Hallo Gemeinde,

mein Catalina System läuft nun nach allerlei Startschwierigkeiten recht gut. Allerdings nur mit deaktivierter [SIP](#). Wenn ich im Clover CsrActiveConfig von 0x67 auf 0x00 setze bootet der Rechner nicht mehr.

Jemand eine Idee ?



```
OpdptioFbDbrv: Starting overrides for \SystemLibrary\CoreServices\boot.efi
Using reloc block: no, hibernate wake: no
ERROR allocating 0x11264 pages at 0x0000000018084000 alloc type 2
Couldn't allocate runtime area
Boot failed: will sleep for 10 seconds before exiting...
-
```

Edit 1:

Wenn ich im Clover Boot screen die Checkbox "Allow unrestricted NVRAM" aktiviere, bootet der Rechner normal.

Im Terminal den [SIP](#) Status abgefragt mit "Csrutil status" ergibt:

Apple Internal: disabled

NVRAM protection: disabled

Rest alles enabled.

Vielleicht kennt jemand die genaueren Zusammenhänge, bzw. hat einen Rat ?

Beitrag von „user232“ vom 30. Oktober 2019, 06:49

AptioMemoryFix.efi und Konsorten ausprobieren, aber jeweils nur einer davon

Beitrag von „macdream“ vom 30. Oktober 2019, 15:09

Hallo user232,

Inwiefern hängt der AptioMemoryFix mit der [SIP](#) zusammen ? Ich habe gerade in alle Richtungen Memoryfixes getestet und war froh dass jetzt alles läuft. Vermutest du einen Zusammenhang oder bist du sicher? Dann fange ich nochmal an...

Von [CMMChris](#) habe ich gelernt, dass der EmuVariableUefi nicht mit AptioMemoryFix zusammen geht. Allerdings habe ich keine Ahnung ob ich diesen (EMU) wirklich brauche, in den EFI Versionen hier im Board für ähnliche Setup's war er immer enthalten.

Beitrag von „CMMChris“ vom 30. Oktober 2019, 15:31

Das habe ich nie gesagt. Andere führen das zwar manchmal an ich weiß aber aus Erfahrung dass das nicht stimmt. AptioMemoryFix funzt aber glaube ich mit deinem Board nicht.

Die [SIP](#) Problematik habe ich nun schon bei diversen z390 Boards beobachtet. Keine Ahnung wie das zusammenhängt.

Beitrag von „macdream“ vom 30. Oktober 2019, 15:49

Hi Chris,

auf jeden Fall habe ich das hier im Forum von einem der Experten gelesen und aufgeschrieben,

allerdings nicht den Author, kann sein das ich dich verwechselt habe.

Sollte dies so sein: Entschuldigung!

Könnte es denn sein das die MemoryFix Treiber für [SIP](#) eine Rolle spielen? Wie kann ich feststellen ob ich den UEFI-EMU überhaupt brauche ?

Beitrag von „CMMChris“ vom 30. Oktober 2019, 15:58

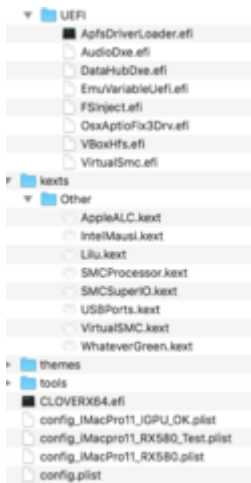
Du hast ein Z390er Board, da funktioniert in der Regel kein nativer NVRAM Zugriff. Entsprechend muss man emulieren. Testen kannst du das indem du den Emu rausnimmst, neustartest, dann eine Variable in den NVRAM setzt, nochmal neustartest und prüfst ob sie noch da ist. Wenn nicht, geht kein nativer NVRAM Zugriff und er muss emuliert werden.

Welchen Aptiofix nutzt du aktuell?

Beitrag von „macdream“ vom 30. Oktober 2019, 18:57

Das klappt selbst mit EMU nicht, habe ich schon probiert, die Variable "mein_name" war nach dem nächsten Booten weg, da aber sonst alles funktionierte habe ich dem Thema keine große Bedeutung zugemessen.

Aktuell boote ich mit:



Nachtrag:

Ich habe jetzt mit OsxAptioFixDrv-free2000.efi getestet:

sudo nvram meine_variable=test

nvram -p gibt (unter anderem) **meine_variable test** aus.

Nach einem Neustart ist die Variable allerdings weg.

Aber: Nun "merkt" Clover sich das letzte Startlaufwerk, das hat bis hierhin noch nie geklappt.

Booten mit [SIP](#) enabled 0x00 klappt auch. 🐞

Es bleibt allerdings eine gewisse Sorge bei dem Einsatz dieses Treibers. 🤔

Ist die Nutzung dieses Treibers weit verbreitet, oder befolgen doch lieber alle den Rat des Entwicklers diesen nicht zu nutzen ?

[CMMChris](#) ich meine, du hättest gute Erfahrungen mit dem Treiber gemacht, hast du ihn "produktiv" im Einsatz oder nur zu Testzwecken genutzt ?

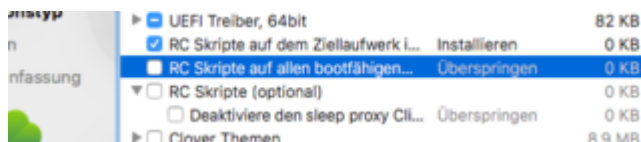
Beitrag von „CMMChris“ vom 30. Oktober 2019, 19:04

Den free2000 sollte man nicht nutzen. Ich persönlich brauche den auch nicht, habe ja kein Z390er Board - würde ich mir auch nie in den Rechner bauen.

Für Emu musst du die RC Skripte installieren sonst bringt er dir nüscht.

Beitrag von „macdream“ vom 30. Oktober 2019, 19:16

Ich nehme an, du meinst diese Skripte:



Hier nehme ich immer die Option ...auf dem Ziellaufwerk installieren...

denn die andere Option ...RC Skripte werden auf allen internen bootfähigen macOS Laufwerken, **außer** dem Ziellaufwerk installiert.... erschließt sich mir nicht.

Beitrag von „Harper Lewis“ vom 30. Oktober 2019, 19:21

Dann schau mal bitte nach, ob die RC Skripte auch installiert wurden:

Spoiler anzeigen

Beitrag von „macdream“ vom 30. Oktober 2019, 19:32

Hallo Harper Lewis,

ich vermute du meinst auf dem Bootlaufwerk /etc

also im Terminal cd /etc

ls -al

...da gibt es nur rc.common und rc.netboot directories und natürlich jede Menge anderes Zeug
Seltsam ?

Beitrag von „Harper Lewis“ vom 30. Oktober 2019, 19:46

Ja, genau - das Startvolumen meinte ich. Dann wurden die Skripte anscheinend nicht dort installiert. ~~Welche Version von macOS nutzt du?~~

Beitrag von „macdream“ vom 30. Oktober 2019, 19:54

Catalina, aber es gibt ja nicht mal die Verzeichnisse. Oder werden die auch von Clover angelegt?

Nachtrag:

Vielleicht installiere ich Clover nochmals und nehme die "seltsame" Option: Skripte überall hin, bloß nicht auf's Startvolumen ?

Beitrag von „cobanramo“ vom 30. Oktober 2019, 20:05

[Zitat von cobanramo](#)

Nur mit **EmuVariableUefi-64.efi** wirds nicht funktionieren, da musst du auch den AptioDriver angucken.

1. **OsxAptioFixDrv** verschiebt Kernel-Regionen, deren gewünschter Speicherbereich blockiert ist temporär und stellt sie nach dem Bootvorgang wieder her. = **NVram nativ Beschreibbar**.
2. **OsxAptioFix2Drv** tut dies nicht (wenn was blockt, ist Schluss).
3. **AptioMemoryFix** = **OsxAptioFix2Drv** + **slide-Automatik** (Speicherbereich des Kernel wird dynamisch festgelegt, beugt Bootfehlern vor) + **"EmuVariableUefi-64.efi"** + weitere kleine Fixes.
4. **OsxAptioFix3Drv** = **OsxAptioFix2Drv** + **"EmuVariableUefi-64.efi"**... warum es den gibt? Weil Slide nicht vorhanden.

Gruss Coban

Edit;

Hier mal ne auszugsweise Originalen;

New set of memory fix drivers.

OsxAptioFixDrv and **OsxAptioFix2Drv** reverted to rev before 4368 so without NVRAM support. Let them stay here for history. And sometimes **OsxAptioFixDrv** is only working version for high warped memory map.

OsxAptioFix2Drv is a reduced version without relocation blocks and may not work on some hardware or with older OS version. But this version supported Hibernation while first one no.

OsxAptioFix3Drv is a **OsxAptioFix2Drv** rev 4369 renamed to be different. This version supported NVRAM. More runtime fixes in 4379. This version can be improved by clover

crew in future.

AptioMemoryFix included in the package is third-party memory fix driver used as is. It is probably better than OsxAptioFix*. Choose it for a first attempt.

OsxLowMemFixDrv is also memory fix driver, not improved since rev608.

SIC! These 5 drivers should not be used simultaneously! Choose one of them!

Clover will be updated for mutually exclusion.

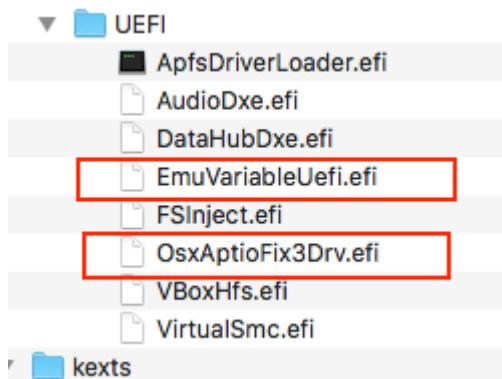
AptioInputFix is included in the package now. It is keyboard driver to support FileVault2 for those who have AMI UEFI BIOS.

Alles anzeigen

Hier mal ein info aus einer meiner älteren Posting's...

Ergo ist das was da drinne hast falsch 😊

EDIT: Das heisst das der **EmuVariableUefi-64.efi** nur mit **OsxAptioFix2Drv** etwas tut, alleine oder mit den anderen zusammen nutzlos ist.



Gruß Coban

Beitrag von „Harper Lewis“ vom 30. Oktober 2019, 20:05

Der Clover-Installer legt die in Beitrag #10 erwähnten Dateien und Verzeichnisse an. Da du Catalina nutzt: Das Startvolume wird schreibgeschützt gemountet, daher muss man das vor dem Starten des Clover-Installers aufheben: `sudo mount -uw /. RC Skripte auf dem Ziellaufwerk installieren` passt.

Beitrag von „macdream“ vom 30. Oktober 2019, 20:13

Hallo Cobanramo,

tut mir leid ich kann dir nicht folgen. Die beiden Treiber hatte ich doch in auch in meiner Config, damit konnte ich mit aktiver [SIP](#) nicht booten.

Was wolltest du mir mit diesem Bild sagen ? Welche Kombi würdest du empfehlen ?

[Harper Lewis](#)

Ok, das macht Sinn. Warum Clover dann keine Fehlermeldung rausgibt ist allerdings seltsam.

Ich werde es mal so versuchen. Danke für den Tip.

Beitrag von „cobanramo“ vom 30. Oktober 2019, 20:17

Zusammengefasst würde das so bedeuten.

Da dein Hack mit OptioFix3Drv und Aktivierter [SIP](#) läuft brauchst du diesen EmuVariableUefi nicht, da du die Clover Skripte nicht oder noch nicht installiert hast funktionierte das ganze noch nicht.

Empfehlen würde ich persönlich immer **AptioMemoryFix** & die Skripte zu installieren.

Gruss Coban

Beitrag von „Harper Lewis“ vom 30. Oktober 2019, 20:20

[Zitat von cobanramo](#)

Empfehlen würde ich persönlich immer **AptioMemoryFix** & die Scripte zu installieren.

Die RC-Skripte benötigt man doch nur, wenn der NVRAM emuliert werden muss, oder habe ich dich da falsch verstanden?

Beitrag von „cobanramo“ vom 30. Oktober 2019, 20:25

Unter anderem ja, sonst funktioniert zumbeispiel der "LastBootetVolume" auch nicht.

Gruss Coban

Beitrag von „macdream“ vom 30. Oktober 2019, 20:25

[Zitat von cobanramo](#)

Zusammengefasst würde das so bedeuten.

Da dein Hack mit OptioFix3Drv und Aktivierter [SIP](#) läuft brauchst du diesen EmuVariableUefi nicht, da du die Clover Scripte nicht oder noch nicht Installiert hast funktionierte das ganze noch nicht.

Empfehlen würde ich persönlich immer **AptioMemoryFix** & die Scripte zu installieren.

Nein, er läuft nur ohne [SIP](#) mit OptioFix3drv und EMU. Wenn ich den free-2000 nehme läuft er auch mit [SIP](#)

Beitrag von „mhaeuser“ vom 30. Oktober 2019, 20:26

Was hier wieder für ein Quatsch geschrieben wird... EmuVariable fungiert absolut unabhängig von jeglichen MemoryFixes und funktioniert auch mit AptioMemoryFix wunderbar. Kein MemoryFix hat *direkt* etwas mit [SIP](#) zu tun, aber [SIP](#)-Optionen könnten theoretisch (das Symptom suggeriert dies, aber ich werde nicht wirklich nachschauen) die Speicherallokierungsmuster von boot.efi beeinflussen, was dann sehr wohl etwas mit den MemoryFixes zu tun hat. Ein Speicherallokierungsproblem liegt nach der Fehlermeldung auch offensichtlich vor.

Man sollte nicht "die MemoryFixes durchprobieren", man sollte *immer* AptioMemoryFix verwenden, wenn es läuft. Wenn Speicherallokierungsfehler auftreten, hier mit beeindruckenden 300(!) MB (erwarten würde ich max. 200 MB), sollte man *nicht* Fix20000 probieren, dieser Treiber ist Müll und kann zu allerhand Instabilitäten führen, sondern diesen hier: <https://github.com/ReddestDream/OcQuirks/>

Der Treiber basiert auf unserem aktuellen OpenCore-Code und enthält einen optionalen (standardmäßig *aus*!) DevirtualiseMmio-Quirk, der auf neuen Systemen (vorallem mit Thunderbolt) diese Probleme dramatisch verringern kann. Der Treiber wird von uns aber *nicht* unterstützt, dafür müsste man schon OpenCore direkt nutzen, wo das ganze über die Config angepasst werden kann.

Beitrag von „macdream“ vom 30. Oktober 2019, 20:37

Hallo Download-Fritz,

vielen Dank für deinen erhellenden Beitrag. Wenn ich dich richtig verstehe, könnte man **FwRuntimeServices.efi** und **OcQuirks.efi (gemeinsam)** als Ersatz für jeglichen AptioMemoryfix nehmen.

Einfach in den Clover Driver Ordner kopieren, richtig ? Irgendwelche Slide=? erforderlich ?

Noch eine Frage: ...wird von uns nicht unterstützt... wer ist mit "uns" gemeint ?

Beitrag von „mhaeuser“ vom 30. Oktober 2019, 20:41

[macdream](#) Genau. "Uns" meint Acidanthera

Beitrag von „cobanramo“ vom 30. Oktober 2019, 20:52

Meine aussage zu falsch ist eigentlich zu dem hier..

OsxAptioFix3Drv = OsxAptioFix2Drv + "EmuVariableUefi-64.efi"

wenn du den OsxAptioFix3Drv. efi drinne hast brauchst du doch keine zusätzliche EmuVariable.. der ist doch mit der Fix3 dabei..

Naja..

Beitrag von „mhaeuser“ vom 30. Oktober 2019, 21:01

Habe ich beim schnellen Lesen wohl falsch verstanden, weil ich das noch nie irgendwo gelesen habe, stimmt aber genauso wenig.

EDIT: Achso, und verlesen habe ich mich offenbar auch nicht:

Zitat

EDIT: Das heisst das der **EmuVariableUefi-64.efi** nur mit **OsxAptioFix2Drv** etwas tut, alleine oder mit den anderen zusammen nutzlos ist.

Beitrag von „cobanramo“ vom 30. Oktober 2019, 21:10

Indem fall bemerke ich das was wir die ganze zeit predigen wohl alles falsch ist.. 😊

Klär uns doch bitte mal auf Fritz was das ganze mit diesen 3-4 verschiedenen Memory Fixe auf sich hat. Was bedeuten die, welcher ist wann zu benutzen, warum gibt es eben die in 3-4 Ausführungen? 😊

Wäre sicher ne grosse hilfe. 😊

Gruss Coban

Beitrag von „mhaeuser“ vom 30. Oktober 2019, 21:19

AptioFix1: Nutzt einen Relocation-Block. Der Kernel wird *nicht* an die von boot.efi angefragte Position kopiert, sondern an eine beliebige, und dann vor dem Kernelstart "zurückkopiert" (an die eigentliche Zielstelle). Kann theoretisch Runtime-Speicher überschreiben und zu unerwarteten Adressen (bis dato kein Problem) in Boot-Args, Device Tree und Konsorten führen. Gehört in den Müll.

AptioFix2: Kein Relocation-Block. Das erfordert, dass die Zielposition frei ist. Gehört in den Müll, siehe nächste Beschreibung.

AptioFix3: AptioFix1 und AptioFix2 verhindern, dass Runtime-Data-Spannen von boot.efi verschoben werden, fixt NVRAM für Aptio 4. AptioFix3 verhindert zusätzlich, dass RT-Code verschoben wird, fixt NVRAM für frühe Aptio 5. Es gibt keine Nachteile gegenüber AptioFix2.

AptioMemoryFix: Puh, es gab diverse Kleinigkeiten gegenüber AptioFix2 (AptioFix3 ist ein Backport von ein paar der AMF-Änderungen gegenüber AF2), an die ich mich nicht mehr erinnere. Die Änderungen, auch wenn ich mich nicht mehr an sie erinnere, sind alle sinnvoll. Bringt geneüber allen anderen AFs eine automatische Slide-Berechnung mit, die auf manchen Systemen aber etwas wackelig ist. Dies kann in sehr, sehr seltenen Fällen zu statistischen Nachteilen (fehlerhafte Bootversuche) gegenüber AF3 führen, aber in diesen Fällen laufen *beide* nicht rund. Wenn AF3 rundläuft, läuft AMF ebenfalls rund, nur, dass es die vorig genannten Änderungen beinhaltet.

OcQuirks: Die Evolution von AptioMemoryFix, ein externer Treiber, der unsere OpenCore-

Bibliothek aufruft. Die Slide-Berechnung wurde korrigiert und "DevirtualiseMmio" hilft sowohl dabei, als auch bei Systemen, die sonst auf free20000 angewiesen wären. Wäre der Treiber nicht "privat" verwaltet (außerhalb von Acidanthera), würde ich diesen bedingungslos empfehlen... so empfehle ich ihn technisch, aber auf eigene Gefahr.

free20000: Pflügt einmal über den Speicher und gibt beliebige (teils wichtige) Regionen einfach frei, nicht der Rede weg, in den Müll.

Beitrag von „cobanramo“ vom 30. Oktober 2019, 21:29

Danke Fritz 😊

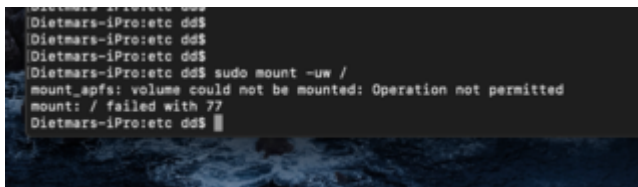
Also Fixt der AptioFix3 auch den Nvram, klar, ist halt auch alt. das der AptioMemoryFix & **FwRuntimeServices.efi** neuer und moderner ist das ist wohl auch klar. 😊

Gruss Coban

Beitrag von „macdream“ vom 30. Oktober 2019, 21:29

[cobanramo](#)

Schreibschutz aufheben wie vorgeschlagen klappt bei Catalina wohl nicht:



```
Dietmars-iPro:etc dd$  
Dietmars-iPro:etc dd$  
Dietmars-iPro:etc dd$  
Dietmars-iPro:etc dd$ sudo mount -uw /  
mount_apfs: volume could not be mounted: Operation not permitted  
mount: / failed with 77  
Dietmars-iPro:etc dd$ █
```

Noch einen anderen Vorschlag ?

Wenn ich von einer anderen Platte starte, müsste Clover doch alle Platten beschreiben können, oder ?

Beitrag von „cobanramo“ vom 30. Oktober 2019, 21:35

Wie meinst du das vom anderen Platte starten? Wenn du das von einem anderen Os aus tust bist du nicht der Besitzer dieser Partition oder Ordner, daher die Fehlermeldung.

Gruss Coban

Beitrag von „macdream“ vom 30. Oktober 2019, 21:38

[mhaeuser](#)

Mann, o Mann, das nenn ich mal eine ausführliche Erklärung, vielen Dank !

[cobanramo](#)

Nein, ich kann Mojave von einer SSD und Catalina von einer anderen starten, und da sind noch ein paar andere Volumes...



so, jetzt 2te Halbzeit Fussball !

Beitrag von „cobanramo“ vom 30. Oktober 2019, 21:43

Aa jetzt.. das ist der [SIP](#) der verhindert, für den müsstest du
Filesystem Protections: disabled haben.

Gruß Coban

Beitrag von „macdream“ vom 30. Oktober 2019, 21:48

also [SIP](#) Abschalten 0x67, dann Clover RC-Scripte installieren und dann [SIP](#) wieder aktivieren.
Zwischendrin noch Schreibschutz mit Mount -uw aufheben?

Beitrag von „cobanramo“ vom 30. Oktober 2019, 21:51

Genau, im Moment ist da die Clover Entwicklung noch nicht so weit denke ich, die kriegen das schon sicherlich irgendwie hin.

[Sip](#) ausschalten, -rw mounten, Clover aktualisieren, [Sip](#) einschalten. rw behebt sich nach neustart.

Gruß Coban

Beitrag von „macdream“ vom 30. Oktober 2019, 21:55

OK, werde ich morgen weiter testen.

Nochmals vielen Dank an euch beide !

Beitrag von „macdream“ vom 31. Oktober 2019, 22:46

Hallo zusammen,

ich habe jetzt nochmals getestet:

Mit AptioMemoryFix und aktiver [SIP](#) klappt das Booten manchmal, meistens jedoch nicht. Wenn ich mit AptioMemoryFix gebootet habe, dann **FwRuntimeServices.efi** und **OcQuirks.efi** in den Clover Ordner kopiere (und den AM lösche) bootet der Rechner mit aktiver [SIP](#) (0x00) nur einmal, dann nicht mehr. Wenn ich im Clover menu "Allow unrestricted NVRAM" aktiviere bootet er mit AM und mit QCQuirks.

Also ich kann mir da keinen Reim darauf machen. Weiß jemand Rat ?

Könnte es mit Mojave einfacher, bzw. zuverlässiger laufen ?

Ich habe an dem Gebastel, anders kann man es wirklich nicht bezeichnen, keine rechte Freude. Man bekommt Vorschläge, dies oder jenes doch bitte einmal zu testen, aber meist weiß man nicht warum.

Mir wäre auch geholfen wenn mir einer sagen könnte, mit deinem Board wird das nie was, weil dies und das. OK, welches Board soll ich nehmen ?

Oder jemand **weiß** welche Maßnahmen zum Erfolg führen, und macht entsprechende Vorschläge, aber diese Ausprobiererei nervt wirklich.

[cobanramo](#)

Das NVRAM scheint zu klappen, meine Testvariable "überlebt" den Neustart! Hilft für mein Bootproblem leider nix...

Beitrag von „cobanramo“ vom 31. Oktober 2019, 23:01

Guck, damit der Hack mit aktivierter [SIP](#) und ohne sorgen laufen kann muss wirklich fast alles

passen,

mit passen mein ich der muss sagen wir mal so salopp 99% dem originale Apple Hardware angenähert sein.

Dafür musst du eben all die problem stellen fixen.

Ich kenne leider deine Hardware nicht aber es gibt sicherlich viele Leute die das schon hinbekommen haben und dir Unterstützung anbieten.

Ansonsten muss man sich halt bissl der Materie einlesen.

Hackintosh ist halt Geduld Sache 😊

Gruss Coban

Beitrag von „macdream“ vom 31. Oktober 2019, 23:17

Ich habe mich natürlich vorher schon mit der Materie befasst, ist auch bereits mein zweiter Hacki, allerdings war das mit HighSierra und anschließendem Umstieg auf Mojave, weitaus einfacher.

Naja, wie gesagt, wenn Mojave zuverlässig läuft, auch gut. Wenn weder Mojave noch Catalina wirklich sicher laufen, kommt das Zeug wieder weg und ein echter Mac her...

Und vermutlich habe ich auch in das falsche Board investiert, wobei mir der Z390 Chipsatz doch weit verbreitet schien hier im Forum, sonst hätte ich nicht auf diesen Chipsatz gesetzt.

Beitrag von „mhaeuser“ vom 1. November 2019, 08:03

[macdream](#) Ich habe zu OcQuirks geraten, wie gesagt, wegen dem DevirtualiseMmio-Quirk, der standardmäßig *de*aktiviert ist. Da wäre es doch naheliegend herauszufinden, wie man ihn aktiviert (an/aus) und konfiguriert (Whitelist), oder? 😊

Beitrag von „macdream“ vom 1. November 2019, 08:25

Guten Morgen,

zunächst fehlt mir die Vorstellung was der DevirtualiseMmio-Quirk bewirkt und ich bin davon ausgegangen, dass dieser aus guten Grund deaktiviert sein wird.

Wenn es für die Speicherallozierung / Verwaltung beim Booten hilft, gerne. Hast du noch einen Tip wo ich suchen muss ?

Ich muss jetzt weg, melde mich später nochmals.

Beitrag von „cobanramo“ vom 1. November 2019, 08:29



OpenCore

Reference Manual (0.5.2)

[2019.10.26]

Seite 15;

5.4 Quirks Properties

- 1. AvoidRuntimeDefrag**
Type: plist boolean
Fallback: false
Description: Protect from boot.efi runtime memory defragmentation.
This option fixes UEFI runtime services (date, time, NVRAM, power control, etc.) support on many firmwares using SMM backing for select services like variable storage. SMM may try to access physical addresses, but they get mowed by boot.efi.
Note: Most but Apple and VMware firmwares need this quirk.
- 2. DevirtualizeMmio**
Type: plist boolean
Fallback: false
Description: Remove runtime attributes from select MMIO regions.
This option reduces stolen memory footprint from the memory map by removing runtime bit for known memory regions. This quirk may result in the increase of KASLR slices available, but is not necessarily compatible with the target board. In general this fixes from 64 to 256 megabytes of memory (present in the debug log), and on some platforms it is the only way to boot macOS, which otherwise fails with allocation error at bootloader stage.
This option is generally useful on all firmwares except some very old ones, like Sandy Bridge. On select firmwares it may require a list of exceptional addresses that still need to get their virtual addresses for proper NVRAM and liberation functioning. Use `MmioWhitelist` section to do this.
- 3. DisableSingleUser**
Type: plist boolean
Fallback: false
Description: Disable single user mode.
This is a security option allowing one to restrict single user mode usage by ignoring CMD+Q hotkey and `-s` boot argument. The behaviour with this quirk enabled is supposed to match TB-based model behaviour. Read [this article](#) to understand how to use single user mode with this quirk enabled.

Gruss Coban

Beitrag von „mhaeuser“ vom 1. November 2019, 10:29

Es ist wegen dem letzten Absatz deaktiviert, weil es nicht garantiert ohne die Whitelist funktioniert, die musst du halt experimentell befüllen. Und jam es besteht eine große Chance, dass der Quirk odentlich hilft. Ich weiß nicht, ob/wie OcQuriks loggt, da biste auf dich alleine gestellt

Beitrag von „redbelt“ vom 1. November 2019, 12:45

Wenn ich [SIP](#) aktiviere mit 0x00 und ich kexte oder Clover aktualisiere, muss ich [SIP](#) dann immer wieder deaktivieren?

Beitrag von „CMMChris“ vom 1. November 2019, 12:46

[SIP](#) hat keine Auswirkung auf die Clover Kext Injection.

Beitrag von „macdream“ vom 1. November 2019, 12:55

Das ist mir auch noch nicht klar, nach meinem Verständnis sorgt die [SIP](#) gerade dafür das keine unsignierten / nicht autorisierten Kexte geladen werden.

Ich war der Meinung, dass dies nur klappt wenn die Kexte aus dem Cache geladen werden.

Oder wie läuft das ?

Beitrag von „CMMChris“ vom 1. November 2019, 13:00

Wenn du die Hackintosh Kexte im System hast musst du [SIP](#) deaktivieren. Wenn du sie mit Clover oder einem anderen Bootloader injectest kann die [SIP](#) aktiviert bleiben.

Beitrag von „redbelt“ vom 1. November 2019, 13:04

Wenn ich [SIP](#) aktiviere, streikt mein System. Beim booten durchgestrichene Kreis. Bestimmt wieder was übersehen.

Beitrag von „CMMChris“ vom 1. November 2019, 13:06

Mach Verbose Boot. Wahrscheinlich "couldn't allocate runtime area". Das Problem haben einige Z390er.

Beitrag von „macdream“ vom 1. November 2019, 13:13

[cobanramo](#)

Ich habe mir jetzt mal die Doku zu OC angesehen, werde gleich mal die empfohlenen [BIOS einstellungen](#) prüfen und den Slide=0 entfernen, den CSM Support kann ich nicht disable, denn dann startet mein Rechner nicht mehr, bzw. die XFX RX580 bringt kein Bild mehr raus.

Aber bezüglich der Whitelist stehe ich auf dem Schlauch, was muss wo eingetragen werden um den DevirtualiseMmio-Quirk zu aktivieren ?

[CMMChris](#)

Ja, genau. Das Problem habe ich dann auch, hängt mit dem Speicher zusammen. Allerdings gelingt es manchmal auch ohne Fehler mit der gleichen Config zu booten.

Das verstehe ich nicht, die Adressen der Speicherbereiche müssten doch nach einem reset immer gleich sein oder ? Da wir doch alles neu initialisiert...

Beitrag von „redbelt“ vom 1. November 2019, 13:15

[CMMChris](#)

Jupp...so ist es. Nix mit aktivieren.

Beitrag von „CMMChris“ vom 1. November 2019, 13:16

[macdream](#) Die Memorymap kann sich durchaus veraendern auch abhaengig von angeschlossenen Geraeten.

Beitrag von „mhaeuser“ vom 1. November 2019, 13:25

[macdream](#) Die Whitelist wird nur relevant, wenn der Quirk zu einem KP führt, also lass es erst mal so weit kommen

Beitrag von „macdream“ vom 1. November 2019, 15:29

[cobanramo](#)

die in #49 genannten Änderungen haben leider nichts gebracht, abgesehen davon, dass ich einmal sogar 2 mal hintereinander mit [SIP](#) booten konnte (man freut sich ja auch über die kleinen Dinge), aber dann wieder nicht mehr. Weder Reset noch Kaltstart, wenn es klappt, dann nur nach Warmstart -> Neustarten vom OS aus.

Also auf "UserEbene" kein Unterschied zu AptioFix oder dem free-2000 Gedöns.

Beitrag von „cobanramo“ vom 1. November 2019, 15:55

Vorschlag;

Nimm den "Debug" OpenCore.efi..

Unter Misc/Debug/Target --> stell den 19 auf 75 und mach mal ein kompletten Start.

Jetzt solltest du in deiner Efi einen ausführlichen Log haben, ne Analyse davon ist sicher nicht verkehrt, lass uns daran mal teilhaben, später kannst du den wieder auf 19 stellen.

Bspl. Kannst du mal versuchen ob dein Rechner den "Safe Mode" Starten kann, wenn nicht sind da schon Grundlegende Sachen noch nicht Komplet richtig eingerichtet...

Abhilfe schafft "EnableSafeModeSlide"

Bspl. Da wäre noch der "EnableWriteUnprotector" oder der "ProvideCustomSlide", eben man müsste sich die Sache mal wirklich angucken und ausarbeiten.

Wenn im Log wirklich alles Sauber abgearbeitet wird seh ich kein Grund warum es nicht funktionieren sollte, so jedenfalls bei mir.

Gruss Coban

Beitrag von „mhaeuser“ vom 1. November 2019, 16:07

Ja, wenn mal alles mögliche befolgt *außer* den Tipp, funktioniert es nicht, das stimmt schon. Wie man den Quirk konfiguriert steht unter der großen Überschrift "Configure OcQuirks" in der README.

Jetzt mit OC rumzuhantieren klingt eher nach zu viel des Guten (wortwörtlich)

Beitrag von „macdream“ vom 1. November 2019, 17:08

[cobanramo](#)

Danke für den gut gemeinten Tip, aber ich denke das ist doch etwas zu hoch für mich. Ich habe in den letzten Wochen auch etliche Posts aus dem Opencore Sammelthread gelesen und bin zu dem Schluss gekommen das dieses Thema eher für die Spezies hier im Board geeignet ist.

Ich habe es an anderer Stelle schon mal gesagt, für mich ist der Hackintosh kein Selbstzweck (Hobby) zum Zeitvertreib, ich beschäftige mich gern mit der Thematik um alles zum Laufen zu bringen, aber dann möchte ich den Hacki auch gerne benutzen, ohne permanent mit Problemen zu kämpfen, oder gespannt zu warten ob die Kiste auch wirklich bootet.

Zur Zeit freue ich mich wenn der Login erscheint, das ist doch nicht normal.

[mhaeuser](#)

Ja, da hast du sicher Recht. Aber ich habe keine Alternative, CSM disable, rote VGA LED auf dem MB an, kein Bild mehr. Allerdings bootet der Rechner und mit ein bisschen Glück kommt man sogar bis in den Login Screen. Also Enabled und die beiden anderen Optionen darunter beide auf UEFI gestellt. Übrigens: Die Grafik bekomme ich nur mit NVRAM reset wieder hin...

Glaubst du das diese Einstellung tatsächlich ursächlich für mein Problem sein könnte ?

Wäre eine ältere Clover Version eine Option, aktuell habe ich 5096 installiert.

Beitrag von „mhaeuser“ vom 1. November 2019, 20:00

Ich glaube, dass das Nichtbefolgen meines DevirtualiseMmio-Tipps, den du wiederholt ignoriert hast, ursächlich für dein Problem ist. Geduld ist eine Ressource...

Beitrag von „macdream“ vom 2. November 2019, 17:02

Tut mir leid, aber wie gesagt, das OpenCore Thema ist mir eine Nummer zu groß. Und ohne CSM support leider kein Bild...

Aber kein Problem, trotzdem Danke für deine Mühe.

[mhaeuser](#)

Nachdem ich nun mehrere Male wirklich auf der Leitung gestanden habe, 🙄 bin ich nun ein paar Schritte weiter. Ich war der Meinung, das es sich bei den von dir erwähnten Parametern um Bootflags in irgendeiner OC config handelt, die ich natürlich nicht habe.

Wie auch immer:

Ich habe nun mit diesen Einstellungen Quirks neu kompiliert (Das habe ich übrigens seit Jahren nicht mehr gemacht)

```
#####  
#                Current Settings                #  
#####  
  
1. QuirksProvideConsoleGopEnable = True  
2. AvoidRuntimeDefrag = True  
3. SetupVirtualMap = True  
4. ProvideCustomSlide = True  
5. DevirtualiseMmio = True  
6. DisableSingleUser = False  
7. DiscardHibernateMap = False  
8. EnableSafeModeSlide = True  
9. ProtectCsmRegion = False  
10. ShrinkMemoryMap = False  
11. ForceExitBootServices = True  
12. DisableVariableWrite = False  
13. EnableWriteUnprotector = True  
  
S. Save  
Q. Quit  
  
Please choose a setting to toggle: █
```

Ich vermute dies war dein Ansinnen, und jetzt mal sehen...

Zunächst nochmals Danke !

Beitrag von „macdream“ vom 9. November 2019, 17:25

Guten Abend zusammen,

Ich denke ich kann nun endlich auch einen Erfolg vermelden, mein Hackintosh läuft mit Catalina 10.15.1, (heute Update gemacht) ohne Probleme.

Es scheint soweit alles zu Laufen:

RX580 über DisplayPort, iCloud, iMessage, AppStore, AppleTv, Sleep, Shutdown, [SIP](#) (dank [mhaeuser](#)) USB3, USB3 dafür nochmals  


Bluetooth USB Dongle, fast immer problemlos (auch nach Neustart), aber manchmal will die Maus nicht. Da hilft nur Batterie raus, oder USB stick kurz rausnehmen.

Gibt es eine in Deutschland erhältliche PCI Karte welche auch Bluetooth-Unterstützung ab BIOS bietet ?

Die separate USB Tastatur unter dem Tisch nervt, genau wie die sporadischen Probleme, wie oben genannt.

Achja, der Samsung EVO 970 Plus traue ich nicht über den Weg, fahre derzeit über eine ältere 128GB SATA SSD und klone von Zeit zu Zeit auf die NVMe.

Manchmal startet die EVO mit Stoppschild...

Ansonsten, fühlt sich fast an wie mein iMac... 

Beitrag von „CMMChris“ vom 9. November 2019, 17:28

Die nativen Apple Karten bieten BIOS Unterstützung in Verbindung mit Apple Peripherie.

Beitrag von „macdream“ vom 9. November 2019, 17:36

Hi Chris,

und diese sind im Handel erhältlich ?

P.S.

Deine Beiträge haben mir über so manches "Problem" hinweg geholfen. Ich habe wieder einiges dazu gelernt.

Macht es aus deiner Sicht Sinn, sich mit OpenCore zu befassen ?

Beitrag von „karacho“ vom 9. November 2019, 18:16

[macdream](#) Sinn macht es immer, sich mit etwas neuem zu befassen. Wenn jedoch mit clover alles zu deiner Zufriedenheit funktioniert, dann nutze ihn doch ruhig weiter. Bei mir lief clover auch tadellos, ich "fummele" jedoch gerne rum, probiere gerne neues aus, und mich hat es einfach nur gereizt OC zum booten zu bewegen. Nun da alles perfekt läuft, möchte ich OC nicht mehr gegen clover tauschen. Vorhin unbeaufsichtigtes Update auf Catalina 15.2 Beta laufen lassen (war in der Küche zum Frikadellen braten) und als ich wieder zum Rechner kam, war er fertig mit der Installation und im sleep. Nach dem Aufwachen angemeldet, die Einrichtung über mich ergehen gelassen und läuft.

Beitrag von „macdream“ vom 9. November 2019, 19:03

Hi [karacho](#),

mir ist nicht klar warum mit OC irgendetwas besser laufen sollte, ist doch "nur" ein Bootloader, das MacOS wird nicht angetastet, die Kexte sind die gleichen,

wo ist der Vorteil ?

Beitrag von „karacho“ vom 9. November 2019, 20:25

[macdream](#) Ich glaube das würdest du nicht fragen, wenn du die Configuration.pdf mal gelesen hättest. Es geht ja auch nicht 'ums besser laufen' mit OC oder Clover. macOS startet mit beiden Loadern gleich gut - bisher jedenfalls - und es gibt auch keinen Unterschied im Betrieb wenn das System erstmal gestartet ist. Sollte Apple jedoch irgendwann (was nicht ausgeschlossen ist und evtl schon mit der nächsten macOS Version kommen 'könnte'), die kextinjection ändern, dann ist Clover wohl erstmal außen vor. OC handelt das nämlich ganz anders. Evtl. könnte [mhaeuser](#) was dazu sagen?

Edit: Oder schau mal hier, hab da mal was raus gesucht...

Multicore
Jun 25, 2016
Discussion
10
This is something I've talked to many developers about before, also Clover developers, formerly Torflex' admin and developer (RIP) and so on. And many of them have had some strange feelings about this, some have not. Throughout the history of macOS and bootloaders, opinions and thoughts have changed.

Here are some reasons why unsigned kexts should be rejected via bootloaders:

- 1. Should and should... This is very appreciated. A reason I don't really want to go on about is a 50.

Expanding via bootloaders secures early boot startup. This means there will be no problems with kexts such as I/O and related kexts. I/O being a kext that is becoming a main load for many things, and will eventually be common standard for a lot, with its plugins etc. Hence the example of I/O. This requires loading before root is mount for panic resolution.

If one wants to `./Library/Extensions` or `/System/Library/Extensions`, it requires rebuilding kext cache, and also ensuring permissions. This is more prone to error.

Stripped kexts in `./Library/Extensions` or `/System/Library/Extensions` are harder to restore. It requires running recovery, mounting boot %s, to analyze and finding the kexts, replacing them, setting permissions, running bootcache. With the bootcache, it is also easier. Doing it via UEFI boot, or other OS, such as Windows, BSD, Linux. As the Mac OS drive is clean, and can be repaired via another system, one could easily restore it that way as well.

If you need to `./Library/Extensions` or `/System/Library/Extensions`, it requires you to disable System Integrity Protection (SIP). Regardless if you ignore security, the kexts expected by Apple. Such systems will be less tested and more error prone.

On a real file, having secure boot (this will soon be with hardware as well using OpenCore), is less possible to install third party using signed kexts, even if they are signed with self-made developer certificates. They can only load one after system startup. This doesn't apply to I/O and many other others. This is for example why a company such as ASUS can't do much, with their drivers for MacOS.

There are of course drawbacks of kext rejection.

Such as...

Injected kexts that do property load dependencies and sometimes don't load at all. This happens more with Clover. OpenCore has files like `it`. Kext rejection can break. Especially for Clover. As Clover relies on a method that is deprecated, and is no longer supported by Apple. However, it might still break for OpenCore. Though OpenCore is made with a different approach, which is more reliable. Injected kexts load some things, i.e. data information through SPI, allowing you to read files in kext directory. This will not work for injected kexts as their dependencies are not mounted. Though, many few kexts need it. In practice, you can restore every kext by itself.

So in this case, injecting in `./Library/Extensions` and `/System/Library/Extensions`, as well as injecting, could be good thing and a bad thing. Especially with Clover. Though the recovery stage could be a clear benefit. For newer methods, in the case of OpenCore, one might want to rely on just injecting.

For me, the "just injecting" approach works well with Clover and OpenCore. But to each their own.

"kext rejection" approach is also called the "vanilla method" by many people. I don't agree with that naming for such method. But calling something vanilla, you refer it to plain, original and clean. The only thing plain, original and clean from the third party's perspective is the drive, when you need it out. Meaning is in the hardware requires the third party's work around, regardless, nothing "vanilla" with it, in my opinion. Though, it is "just rejected" method. Instead using "vanilla install" or "vanilla method" is better and clear.