

Erledigt

Ozmosis Patch macOS Mojave

Beitrag von „kuckkuck“ vom 8. Juni 2018, 19:23

Guten Tag zusammen 😊

Wer nutzt denn hier noch Ozmosis und hat Zugriff auf macOS Mojave? Ich bräuchte ein Versuchskaninchen, am besten jemand der bereits DBounce benutzt hat... [@derHackfan](#) vielleicht?

Ich habe die 167X Version von Ozmosis soweit möglich überarbeitet um Kompatibilität mit Mojave herzustellen. Sollte das ganze funktionieren hätte es sehr gute Zukunftsaussichten, solange sich an macOS nichts fundamentales ändert... Kompatibilität mit neuen OS wäre durch einen einzigen Plist Eintrag herzustellen.

Naja aber soweit ist das alles nur Theorie... Leider fehlt mir aktuell eine Festplatte und die Zeit um das ganze selber zu testen 😞

Beitrag von „Noir0SX“ vom 8. Juni 2018, 20:02

Kann ich leider nicht helfen habe nix was Ozmosis unterstützt, obwohl das schonmal reizt.

Beitrag von „derHackfan“ vom 8. Juni 2018, 21:24

Was hat du denn an den Ozmosis Files verändert, wenn ich dazu komme und oder Zeit finde werde ich es gerne testen, bis jetzt weiß ich nur dass das UnSolid Kext ausgespielt hat.

Beitrag von „kuckkuck“ vom 8. Juni 2018, 23:34

[Zitat von derHackfan](#)

Was hat du denn an den Ozmosis Files verändert

Naja, wie genau möchtest du es haben?

Kurzfassung: Ich habe teile von Ozmosis disassembled und angepasst um zu versuchen die KextInjection unter Mojave zu fixen...

Mittelfassung: Um Kexts aus der EFI zu injecten sind prinzipiell zwei Dinge notwendig:

- readBooterExtensions --> Lesen der Extensions vom devicetree
- loadExecutable --> Umgehen der [SIP](#) Beschränkungen

Die Art wie die einzelnen Befehle ausgeführt/injected werden unterscheidet sich zwischen den Bootloadern.

Wen es interessiert, bei Ozmosis sieht dies so aus:

Spoiler anzeigen

Die AppleKernel haben sich im Laufe der Versionen verändert und so verändern sich auch die Muster nachdem die im Bootloader eingebauten Kernelpatcher suchen um die Befehle zu erlauben.

Unter Clover kann man sich das ganze zB hier in der Source ansehen: <https://sourceforge.net/p/clover/patches/attachment/1595/>

Unterschieden wird hier wieder zwischen readBooterExtensions (EXT) und loadExecutable ([SIP](#)). Clover geht an dieser Stelle einen Dirty Weg und patcht im readStartupExtensions Bereich und überspringt durch einen Hardcode somit den eigentlich vorhergesehenen readPrelinkedExtensions Bereich für das ausführen von readBooterExtensions.

Ok, davon abgesehen sieht man in der oben verlinkten Liste sehr schön in welchen Bereichen ([SIP](#) oder EXT) Veränderungen notwendig waren um die KextInjection zu erlauben. Von 10.10 bis 10.13 veränderte sich an dem EXT procedure nichts, jedoch im [SIP](#) Bereich. Die Veränderungen waren meist marginal und cecekpawon setzte diese in einem disassemble von Ozm um, durch kleine Anpassungen der Assembler-Befehlsreferenzen und hinterlegten Hex-Werte.

An dem [SIP](#) Patch hat sich für Mojave nichts verändert, jedoch am EXT Patch.

Dementsprechend kann das problem nicht wie bisher durch das verändern von einzelnen Bytes im loadExecutable-Bereich behoben werden. Praktisch niemand der in den letzten paar Jahren kleinere Ozm Patches rausgebracht hat besitzt oder hat Zugriff auf die Ozmosis Source. Ozm behandelt das ganze Thema dynamisch, angelehnt an [meklort](#), was jedoch einige Probleme mit sich bringt wenn die Source nicht vorhanden ist und man versucht etwas zu patchen. Für 10.14 musste sich also was anderes ausgedacht werden und somit habe ich zusammen mit cecekpawon ein neues Konzept entworfen, basierend auf ceceks UEFTW Tools und vorallem dem KernnextPatcher.

Dieses neue Konzept sieht in diesem ersten Versuch so aus, dass alte hardcoded Patches von cecek wieder entfernt wurden, der dynamische Kernelpatch Mechanismus von Ozmosis aber trotzdem teilweise deaktiviert wurde und der seit El Capitan nötige und von cecek entwickelte SMBIOS fix wieder in Ozmosis.efi eingefügt wurde. Die nötigen Kernelpatches werden vom Kernnextpatcher übernommen, der über eine Plist, angelehnt an die in Clover hinterlegten Patches, seine Informationen bekommt. In manchen Teilen ist somit die KernelPatch Prozedur jetzt näher angelehnt an Clover, leider weniger dynamisch, aber trotzdem funktionell (siehe Clover)

Man kann das ganze jetzt natürlich noch ausführliche beschreiben, welche Mechanismen innerhalb Ozmosis konkret deaktiviert werden, wo die Unterschiede zwischen dem erzwungenen readBooterExtensions und dem ganzen über den readPrelinkedExtensions sind, aber das würde jetzt sehr einfach nur technischer werden, das wichtigste ist eigentlich gesagt... Ansonsten einfach Fragen 😊

[@derHackfan](#) hast du aktuell ein System mit DBounce bereit? Hast du eventuell sogar schon einen Mojave stick? 😊

Beitrag von „derHackfan“ vom 9. Juni 2018, 18:41

Zur Zeit habe ich es nur geschafft die Beta Installer App aus dem App Store zu laden, sobald mir die Familie Zeit lässt werde ich möglichst viele Optionen testen, am liebsten auf einer frischen SSD ab Werk. 😄

Beitrag von „kuckkuck“ vom 10. Juni 2018, 12:32

Neue Ozmosis.efi:

hackintosh-forum.de/attachment/74780/

Neueste Version des [Kernextpatchers](#):

hackintosh-forum.de/attachment/74779/

KernextPatcher.plist mit den notwendigen KernelPatches für Kext-Injection unter Mojave in Kombination mit der neuen Ozmosis.efi:

hackintosh-forum.de/attachment/74778/ (Wird auf der [EFI für den Patcher abgelegt](#))

- EnhancedFat muss im BIOS das Filesystem ersetzen.
- Alle anderen Treiber (hackintosh-forum.de/attachment/74781/, HFSPlus, Ozmosis, KernextPatcher (...)) sollten [mit DBounce von der EFI](#) geladen werden.
- Eine Shell kann [unter EFI/Boot als BootX64.efi](#) hinterlegt werden um DBounce mit bcfg einzubinden.
- Ebenfalls muss eine Defaults.plist mit den neuesten FirmwareFeatures, FirmwareFeaturesMask, Bios Date und Bios Version in der EFI Ozm Ordnerstruktur vorhanden sein.
- Auch FakeSMC und eventuell weiter notwendiges wie USBInjectAll auf die EFI Partition. (Wie gewohnt nach Efi/Oz/Darwin/Extensions/Common)
- Ist in der defaults das UserInterface aktiviert, braucht es ein [Theme in EFI/Oz](#)

Über Feedback freue ich mich natürlich!

Edit: Release: [macOS Mojave Ozmosis BETA – OZ167X-XMAX](#)

Beitrag von „kuckkuck“ vom 13. Juni 2018, 01:56

Ich hatte die Zeit ein wenig mit Sierra zu testen. Der auf die oben angehängte Ozm Version angewandte Patch bezieht sich nicht nur auf Mojave, sondern auf die generelle Funktionsweise von Ozmosis.

Dementsprechend kann ein Test mit jeglichem Betriebssystem schonmal die allgemeine Funktionsweise überprüfen.

Und was soll ich sagen, die Anpassungen scheinen wie erhofft gewirkt zu haben. Teile der veralteten Ozmosis Patching Prozeduren werden nicht mehr angewandt und bdmesg verändert sich entsprechend.

Spoiler anzeigen

Zudem übernimmt der Kernnextpatcher wunderbar die Aufgabe den Kernel mit neuen Daten zu patchen und sobald dies getan ist springt Ozmosis wieder ein und kann erfolgreich Kexts einbringen.

Obwohl die Herangehensweise also eine neue ist, funktioniert Kext-Injection wunderbar und zuverlässig mit Kexts aus dem ROM und auch aus der EFI.

Deaktiviert man hingegen die KernelPatches des KernnextPatcher und lässt Ozm somit ins Messer laufen, funktioniert auch keinerlei Kext-Injection und der Bootvorgang bleibt bei den bekannten SMC-Issues stecken.

Das ganze beweist also, dass das allgemeine Prinzip und die Idee dahinter funktionieren. Habe ich nicht eine weitere Neuerung von Mojave übersehen, würde das heißen, dass mit dieser Ozm Version die Kext-Injection unter Mojave laufen sollte, Ozm also somit kompatibel wäre. Vielleicht könnte das ja jemand testen und eventuell sogar bestätigen...

Übrigens die Sache mit DBounce ist nur ein Vorschlag, ich habe ein ROM mit den neuen Dateien bestückt und getestet, funktioniert fabelhaft auch aus dem ROM heraus. Das ROM braucht nur genug Platz, was bei meine Z97-D3H jedoch der Fall war...

Beitrag von „kuckkuck“ vom 19. Juni 2018, 22:19

Tests sind erfolgreich abgeschlossen, mit cecekpawon alles besprochen und release vorraussichtlich noch heute.

Hier ein paar Anhänge:



BDMESG

kextstat

KernextPatcher

Beitrag von „derHackfan“ vom 19. Juni 2018, 23:16

Heißt das jetzt ohne KernextPatcher kein Boot mit Ozmosis?

Beitrag von „kuckkuck“ vom 19. Juni 2018, 23:22

Boot schon, aber keine KextInjection.

Egal ob Clover oder Ozmosis, jeder "Bootloader" muss den Kernel patchen um die eingebauten Sicherheitsvorkehrungen zu umgehen und Kexts in den Prelinked Kernel injecten zu können. Dafür haben die Loader eingebaute Patching-Mechanismen.

Die Kernel der macOS Systeme ändern sich und so ändern sich auch die nötigen patches.

Der eingebaute Patching-Mechanismus von Ozmosis hat mit Mojave nichts mehr bewirken können (aufgrund der Kernel-Veränderungen). Da er ohne den Source-Code in diesem speziellen Fall schwer zu patchen ist, haben wir uns dazu entschieden den Mechanismus zu deaktivieren und stattdessen den Kernnext-Patcher den Dienst machen zu lassen. Das hat aber andererseits den Vorteil, dass ab jetzt Kernel-Veränderungen ganz easy im KernnextPatcher eingetragen und somit volle Upgrade-Kompatibilität hergestellt werden kann.

Ein Platz-Problem im Rom sollte es eigentlich nicht geben, da APFS jetzt praktisch komplett wegfällt. Gleiches gilt für die meisten alten Treiber. Nötig ist nur noch der APFSDriverLoader, EnhancedFat (anstatt Filesystem), Ozmosis selber und der KernnextPatcher.

Mein ElCap-Mojave kompatibles System sieht so aus:

```
HfsPlus
APFSDriverLoader
EnhancedFat
Ozmosis
OzmosisDefaults
KernnextPatcher
AcpiPatcher
FakeSMC.Rev-6.26-344-g1cf53906.1787
IntelMausiEthernet.Rev-2.3.0
```

Beitrag von „griven“ vom 19. Juni 2018, 23:25

Ich teste das mal mit meinem ROM ebenfalls...

[@kuckkuck](#) hast Du die Files zufällig gerade als .ffs parat ?!?

Beitrag von „kuckkuck“ vom 19. Juni 2018, 23:28

Erstmal für euch die Finalen Dateien an die, die testen wollen:

[Ozmosis-XMAX.zip](#) mit den neuen Patches

[KernnextPatcher.ffs.zip](#) inklusive eingebauter Plist für Kext-Injection von Mountain Lion --> Mojave (sowohl früheres als auch neueres lässt sich hier ganz leicht noch einbauen)

[APFS.zip](#) loader mit 5 KB (neueste Version 1.1.1)

Edit: Release: [macOS Mojave Ozmosis BETA – OZ167X-XMAX](#)

Beitrag von „derHackfan“ vom 19. Juni 2018, 23:29

Oha, da werde ich dann aber in Zukunft einen Bogen um Ozmosis machen, Hühner aus Bodenhaltung ist nicht mein Ding. 😊

Beitrag von „kuckkuck“ vom 19. Juni 2018, 23:33

Noch eine Sache zu APFS und Ozmosis-Booteinträgen:

Ich kann hier noch kein allgemeines Urteil fällen, aber derzeit sieht es zumindest folgendermaßen aus:

Früher war ein Booteintrag unbeständig, teils bei Reboots, teils nur bei NVRam Resets ist er verschwunden --> bcfg boot add nötig.

Bootet man in die Recovery oder in Sierra oder ähnliches und wählt hier in den Systemeinstellungen als Startvolume ein installiertes Mojave aus, wird vom OS (also kein bcfg) ein Booteintrag erstellt. Nach einigen Tests siehts so aus als wäre dieser Booteintrag bestehend. Keine reboots entfernen ihn und selbst nach NVRam Resets bleibt dieser Booteintrag – bei der Nutzung des neuesten Ozmosis und des APFSDriverLoaders – bestehen.

Beitrag von „al6042“ vom 19. Juni 2018, 23:36

Das ist doch mal eine interessante Entwicklung.

Beitrag von „griven“ vom 20. Juni 2018, 00:03

Vielleicht bin ich inzwischen auch einfach zu doof dafür aber bei mir hat es **NICHT** geklappt. Ich habe die Files eingebaut und das Ergebnis ist das APFS Volumes nicht gefunden werden sprich der APFSLoader wohl nicht greift. Hier mal mein ROM vielleicht habe ich ja was falsch gemacht: hackintosh-forum.de/attachment/75427/

Edit: Defekter Anhang entfernt

Beitrag von „kuckkuck“ vom 20. Juni 2018, 00:12

Bevor ich das mir gleich anschau, führ mal bitte `drivers -b` in der shell aus und schick den output. Hast du mal versucht wie beschrieben einen Booteintrag zu erstellen indem du über einen funktionierenden macOS Boot das Startvolume auswählst?

Kann gerade auf den ersten Blick kein Problem mit dem ROM feststellen, ich muss nur sagen, dass da gefühlt einiges unbenutztes drinnen liegt 😊 (Disabler, PartitionDXE, Extfs etc.)

Beitrag von „griven“ vom 20. Juni 2018, 00:34

Ach Platz ist da ja genug...

Beitrag von „kuckkuck“ vom 20. Juni 2018, 00:41

Sieht eigentlich soweit gut aus... Wenn du ins BIOS gehst und dort zB auf die Auswahl einen Flash-files gehst, müsstest du eigentlich auch auf APFS Platten navigieren können. Das würde schonmal zeigen, dass APFS geladen wird...

Ist aktuell eine Platte mit Mojave oder HS verbunden? Hast du die Startvolume-Auswahl mal probiert?

Beitrag von „griven“ vom 20. Juni 2018, 00:59

~~Das ist der Fall aber ich sehe die APFS Volumes weder im Bios noch auf der Shell...~~

Ich sehe die Volumes sowohl im Bios als auch in der Shell. Im Bios sind es einfach Einträge für die EVO...

Wie auch immer OZ verweigert sich bei mir komplett. Ich kann über Clover starten und eine Startvolume wählen was mir nachher auch im Bios Bootmenu angezeigt wird. Wähle ich das Volume werden die Screens schwarz und das war es dann auch...

Beitrag von „kuckkuck“ vom 20. Juni 2018, 08:55

Mach mal einen NVRam reset und stelle sicher, dass XMAX korrekt geladen wird und deaktivier dann die iGPU im BIOS...

Beitrag von „griven“ vom 20. Juni 2018, 21:27

Habe ich erledigt. OZ wird richtig geladen (HermitCrabs Logo und OZ richtet sich ein) ebenfalls wird der Eintrag für die Recovery Partition von ElCapitan (ist noch irgend was altes) erzeugt. In

der EFI Shell sind alle APFS Partitionen ersichtlich allerdings lässt sich macOS nicht starten. Versuche ich die boot.efi zu starten wird der Bildschirm schwarz (Signal geht nicht verloren sondern es wird einfach nix angezeigt) und auch sonst passiert nichts weiter. Der Inject der OZ Defaults muss aber funktionieren denn ich habe in den defaults den verbose aktiv und boote ich dann mit Clover erfolgt dieser boot im Verbose Mode obwohl Clover hier nichts der gleichen aktiv hat in der config.

Ich habe mal die .plist aus dem KernExt Patcher rausgezogen und das Logging aktiviert folgendes kommt dabei raus:

Code

```
1. 00:100          (00:100)          |          Running          from:
   MemoryMapped(0xB,0xCA2B8004,0xCAB9F003)FvFile(99665243-5AED-4D57-92AF-
   8C785FBC7558)
2. 00:101 (00:001) | SelfDirPath = \EFI
3. 00:108 (00:007) | Get Plist from: NVRAM ... Not Found
4. 00:109          (00:000)          |          Get          Plist          from:          ESP:
   PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x0,0xFFFF,0x0)/HD(1,GPT,CCF2903D-6363-4D64-86E2-
   32ADF75A91C2,0x28,0x64000)\EFI\KernextPatcher.plist ... Success
5. 00:110 (00:001) | Parsing plist: ... Success
6. 00:110 (00:000) | KernelToPatch: 6 requested
7. 00:110 (00:000) | - [00]: KbePMLMav-EXT | MatchOS: 10.8,10.9 | Len: 11 |
   C6E830000000EB084889DF -> C6E83000000090904889DF
8. 00:110 (00:000) | - [01]: KbePYosECSieHS-EXT | MatchOS: 10.10-10.13 | Len: 8 |
   E825000000EB05E8 -> E8250000009090E8
9. 00:110 (00:000) | - [02]: KbePMoja-EXT | MatchOS: 10.14 | Len: 8 | E8AF000000EB05E8 -
   > E8AF0000009090E8
10. 00:110 (00:000) | - [03]: KbePEC-SIP | MatchOS: 10.11 | Len: 16 |
   C34885DB7470488B034889DFFF502848 -> C34885DBEB12488B034889DFFF502848
11. 00:110 (00:000) | - [04]: KbePSie-SIP | MatchOS: 10.12 | Len: 16 |
   C34885DB7471488B034889DFFF502848 -> C34885DBEB12488B034889DFFF502848
12. 00:110 (00:000) | - [05]: KbeHSMoja-SIP | MatchOS: 10.13,10.14 | Len: 16 |
   C34885DB7469488B034889DFFF502848 -> C34885DBEB12488B034889DFFF502848
13. 13:250 (13:140) | Found Booter (OS: 10.12 | Ver: 361.200.9)
14. 13:250          (00:000)          |          Booter          Path:
   PciRoot(0x0)/Pci(0x1F,0x2)/Sata(0x0,0xFFFF,0x0)/HD(2,GPT,2C761CAA-CD36-401D-9B6C-
   C1EEBA520A02,0x64028,0x3A321FE0)/VenMedia(BE74FCF7-0B7C-49F3-9147-
   01F4042E6842,FC255BDB313A2F40BDA3FE2AA788C099)\3C199D35-CB13-3164-9BF0-
   22821813CB91\System\Library\CoreServices\boot.efi
```

15. 13:268 (00:018) | OSVersion: 10.14 | BuildVersion: 18A314h
16. 13:268 (00:000) | KernelAndKextPatcherInit: Start
17. 13:268 (00:000) | ScreenInfo: Base 0xE0000000 Width 1920 Height 1080 RowBytes 0x1E00
18. 13:268 (00:000) | AllocatePages: Got RelocBase (0x5E92000) + valid KernelData (0x6092000)
19. 13:268 (00:000) | RelocBase = 5E92000
20. 13:269 (00:000) | Found BootArgs at 0xC41A000
21. 13:269 (00:000) | Invalid 64Bit kernel at 0xB092000 - skipping patches!KernelAndKextPatcherInit: End
22. 13:269 (00:000) | KernextPatcher: End

Alles anzeigen

Ich denke das könnte eine Erklärung dafür sein das es nicht geht
13:269 (00:000) | Invalid 64Bit kernel at 0xB092000 - skipping
patches!KernelAndKextPatcherInit: End oder was denkst Du?

Beitrag von „kuckkuck“ vom 20. Juni 2018, 23:03

Wie hast du denn das Log gezogen? Über dir Shell?

Es könnte theoretisch schon sein, dass hier ein Problem besteht, aber ich glaube nicht, dass der Kernext den Mojave Kernel wirklich nicht erkennen kann. Es handelt sich hier praktisch um die gleichen Vorgänge wie bei Clover...

Jedoch bist du nicht der einzige mit dem Blackscreen Problem. Sowohl Ich als auch ammoune hatten das gleiche Problem und fanden vorerst keine Lösung. Letztendlich lies sich aber bei uns beiden das Problem ausfindig machen und beseitigen.

In meinem Fall reichte es die iGPU im BIOS komplett zu deaktivieren und schon ging alles problemlos. Ammoune, der die iGPU als einzige GPU benutzt, musste ein wenig seine [BIOS settings](#) anpassen und danach ging alles wie geschmiert. Hier seine Nachricht dazu:

"I've found the trick, need to set in the [bios settings](#):

Intel Processor Graphics Memory Allocation to 32M
[DVMT](#) Total Memory Size to 128M"

Gäbe es hier wirklich ein Problem mit dem Kernel sollte eigentlich trotzdem der Verbose Output starten (solange der Kernel halbwegs funktioniert) und nicht einfach nur ein Blackscreen oder ein unbewegliches Apple Logo erscheinen. Ebenfalls zu solchen Blackscreens können "unverträgliche" Dateien im EFI wie Kexts mit Plugins Folder führen...

Beitrag von „griven“ vom 20. Juni 2018, 23:14

Es macht keinen Unterschied ob ich die Intel im Bios aktiv habe oder nicht das Verhalten ist komplett identisch...

Extensions auf der EFI kann ich ausschließen ist aktuell nichts drin und im ROM ist aktuell auch nur noch die FakeSMC enthalten und sonst nichts weiter...




Der KernExtPatcher schreibt das Log auf die EFI ist dann da im Stammordner zu finden...

Beitrag von „kuckkuck“ vom 21. Juni 2018, 13:25

Seht kryptisch... Ich denke ich bau dir im Laufe des Tages nochmal ein neues Rom, vielleicht hilft's... Hattest du bisher mal Probleme mit zB den MATS Tables? Eine Idee wäre noch den ACPIPatcher zum Header fixen einzubauen...

Zitat von griven

Der KernExtPatcher schreibt das Log auf die EFI ist dann da im Stammordner zu finden...

Das ist schon klar, die Frage ist eher wie du an das Log gekommen  bist. Der Patcher reagiert ja auf das Starten eines Betriebssystems. Wenn du jetzt über den Start der Shell das Log gezogen hast, würde das erkläre warum ein "Invalid Kernel" erkannt wird. Würde mich ja irgendwie echt wundern wenn der KernelPatcher hier das Problem ist, in 3 verschiedenen Fällen hat jetzt die Mojave Installation bei verschiedenen Leuten schon geklappt, der Kernext hatte nie ein Problem  

Ich schau mir das heute abend nochmal an und würde das Mojave-Paket mal heute noch im Forum an Tester releasen, damit es ein breiteres Hardware-Versuchs-Spektrum für Tests gibt. Die Zuverlässigkeit mit der hier Mojave werkelt zaubert mir immer wieder ein Lächeln ins Gesicht 👍

Beitrag von „griven“ vom 21. Juni 2018, 15:20

Nope keine MATS Probleme gibt es bei meinem Board auch gar nicht... Die Idee das es vielleicht auch am ACPI liegen könnte ist mir aber auch schon in den Sinn gekommen. Die DSDT ist in meinem Fall unangetastet.

Beitrag von „kuckkuck“ vom 21. Juni 2018, 20:54

Ich habe dir mal ein neues ROM gebastelt, probiers doch mal als Versuch hiermit:
hackintosh-forum.de/attachment/75617/

In dem Ganzen werkelt jetzt auch der ACPIPatcher mit ein paar Standard Patches. Leider kenn ich deine DSDT nicht so genau, um das perfekt anzupassen, aber die im ROM hinterlegten Patches entsprechen denen aus der angehängten Plist.

Edit: Defekter Anhang entfernt

Beitrag von „griven“ vom 21. Juni 2018, 23:30

So ich habe nun noch mal fleissig getestet und muss ernüchert feststellen es geht nicht bzw. mit meinem Setup geht es nicht...

Das Maximum was ich erreichen konnte war einmal ein Boot in die alte ElCapitan Recovery
[Bios Settings](#)

: IGPU OFF, CSM AN, OS Win8 WHQL, Secure Boot OFF und OZ Defaults DisableAtiInjection=true, DisableIntelInjection=true. Der Versuch Mojave über die Shell mittels boot.efi von FS4 zu starten (FS4::, cd 3C199D35-CB13-3164-9BF0-22821813CB91\System\Library\CoreServices\, boot.efi) wurde mit einem reboot quittiert ohne das verbose irgendeine Chance hatte auch nur irgendwas anzuzeigen. Leider bin ich einigermaßen Ratlos wo es noch hängt. Aber eines ist für mich aber auch schon ein no go denn meine AMD R9-290X spielt mit CSM aktiv nicht wie sie soll und die IGPU dauerhaft deaktivieren ist auch keine wirkliche Option.

Beitrag von „kuckkuck“ vom 23. Juni 2018, 13:16

[Zitat von griven](#)

Ich habe die Files eingebaut und das Ergebnis ist das APFS Volumes nicht gefunden werden

Kleine Neuentwicklung: <https://www.insanelymac.com/fo...ndComment&comment=2621884>

[AppleLoadImage](#) sollte im nächsten Release auch zu finden sein.

Ich habe komischerweise auch ohne AppleLoadImage keine Problem mit APFS bootentries, aber werds trotzdem mal probieren...

Beitrag von „griven“ vom 23. Juni 2018, 13:31

Würde in dem Fall aber bedeuten man nutzt AppleLoadImage und AppleAPFSJumpstart naja man kann es testen aber ich fürchte mein Problem ist in der Hauptsache der NVRAM bei dem Board. Mit dem AptioMemoryFix und Clover funzt der NVRAM wie er soll leider lässt sich das aber nicht auf OZ übertragen denn der AptioMemoryFix tut da eben nicht so wie er soll...

Beitrag von „kuckkuck“ vom 4. August 2018, 00:02

Habe es inzwischen hinbekommen, Mojave auch mit aktivierter und head-less iGPU zu starten, HWEncoding funktioniert wunderbar 😊

Beitrag von „kuckkuck“ vom 20. August 2018, 20:51

Noch eine Sache zu Ozm ROMs, könnte man das vielleicht noch anpassen? [OZMOSIS REQUEST THREAD / Ozmosis Anfragen \(nur bis Haswell B/H/Z97\)](#)

Ich fände

6. erwünschtes OS

noch eine gute Sache, dann weiß man welche Dateien man am besten einbaut (APFS, Injector, PartitionDXE, XMAX, Kernext...)

Das wäre dann:

1. Hersteller:
2. Modell:
3. Chipsatz:
4. Bios Version:
5. Bios Link:
6. erwünschtes OS:

Kann das jemand vielleicht so anpassen oder [@griven](#) seinen Post bearbeiten? 😊

Beitrag von „Coaster“ vom 20. August 2018, 21:03

Ich war mal so frech und habe den Beitrag bearbeitet.

Beitrag von „griven“ vom 26. August 2018, 02:20

Ja tue ich noch allerdings war ich selbst bis Freitag ausser Landes 😊

Beitrag von „kuckkuck“ vom 25. September 2018, 15:26

Ich komme morgen zurück nach Hause und will einige Tests (unter anderem mit Darboot) machen. Den für Mojave nötigen Kernnextpatcher will ich auf den neuesten Stand bringen. Des weiteren gibt es Verbesserungen beim Dissassamble von Ozmosis, mit einigem an Glück und Zeit kann ich vielleicht die internen Patching Mechanismen von Ozmosis wieder zum laufen kriegen. Will jetzt hier aber nichts versprechen, ist eher unwahrscheinlich, da der Grund für das Versagen des alten Patchers noch nicht ganz geklärt ist.

Wenn ich ein FFS Paket für Mojave zusammenschuster, würde mir jemand helfen die Ozmosis Datenbank (zumindest in Teilen) auf den neuesten Stand zu bringen?

Beitrag von „kuckkuck“ vom 27. September 2018, 16:18

Ich habe 4 Pakete zusammengestellt, die unterschiedlich groß sind und auf die verschiedenen ROMs passen sollten:

[Min.zip](#) Minimal-Konfiguration, klein aber etwas unzuverlässiger (kein DarBoot, kein ACPIPatcher für zB MATS Header)

[Mid.zip](#) Mittleres Paket, zuverlässiger (Darboot, Theme und ACPIPatcher), kein SchnickSchnack

[Mid +.zip](#) Erweitertes Paket, vollständig mit etwas SchnickSchnack (ExtFS, AppleBootPolicy, PartitionDXE, FakeSMC Sensoren bis auf GPUSensor)

[Max.zip](#) Größtes Paket für große ROMs, inklusive HermitShell, aller Sensoren und obigem

SchnickSchnack

Die einzelnen Treiber sind nummeriert, damit sie von OZMTool in der richtigen Reihenfolge injected werden. Das Updaten der ROMs geht damit ziemlich schnell, ich habe auch bereits einige geupdated...

Beitrag von „derHackfan“ vom 28. Oktober 2018, 22:10

[Zitat von kuckkuck](#)

Das Updaten der ROMs geht damit ziemlich schnell

Ja, aber funktionieren sie auch auf Anhieb? 😊

Ich habe mit meinem ASRock Z77 Pro4 angefangen und um Rückmeldung gebeten, anschließend folgt Z87, Z97 und H81M mit DBounce, es wäre erfreulich wenn die Möhren booten ohne weitere Anpassungen.

Beitrag von „kuckkuck“ vom 29. Oktober 2018, 00:33

Sollte eigentlich der Fall sein... Was hast du denn alles eingebaut? (Ich kanns mir später auch selber nochmal anschauen)

Beitrag von „derHackfan“ vom 17. Januar 2019, 08:46

[Zitat von kuckkuck](#)

Ich weiß nicht, es ist komisch, bei vielen funktioniert auch einfach und dann gibt es diese komischen Fälle, bei denen irgendwie angeschlossene Speichermedien das UEFI zum Einfrieren bringen... Manchmal löst sich das Problem auf absurde Weise, manchmal funktioniert eine ältere BIOS-Version und manchmal geht es pauschal einfach nicht.

[kuckkuck](#) Guten Morgen,

ich bin damit mal nach Intern gewandert, weil das muss ja nicht jeder mitlesen. 😊

Nach dem gestrigen Versuch das Mini und oder das Mid-Paket auf meinem Z77 zu flashen hätte es mich fast das Board gekostet, ein ahnungsloser User ohne USB-Programmer hätte jetzt einen Briefbeschwerer.

Irgend ein FFS sorgt dafür, dass sprichwörtlich nichts mehr geht, man kann am Ende nicht mal mehr ein original [BIOS flashen](#), daher bin ich der Meinung, dass man die User unbedingt vorher ausdrücklich darauf hinweisen muss, dass es ggf. arge Probleme gibt.

Bei ASRock hat man teilweise nicht mal die Möglichkeit auf ein älteres BIOS auszuweichen, unter Support findet man nur ein aktuelles und ein BETA BIOS mehr nicht, wie das bei den anderen Herstellern ist, kann ich nicht sagen.

Mal davon abgesehen, dass sich das original ASRock BIOS auch nur sehr widerwillig mit dem USB-Programmer flashen ließ und bis die Möhre wieder lief, war es Mitternacht. 😞

Aber letzteres ist ja ein anderes Problem...

Gruß

Beitrag von „kuckkuck“ vom 17. Januar 2019, 12:56

Ist ja echt interessant, dass du so arge Probleme hast... Wieso hast du denn Min/Min erneut geflasht, wenn es das erste mal schon Probleme gab? 😞

Irgendwie wundert es mich, dass ich noch nie hier gelesen habe, dass jemand wirklich einen Briefbeschwerer wegen XMAX erhalten hat. Das in manchen Fällen ein unerklärliches Problem besteht ist mir aber natürlich klar. Leider konnte ich selber noch nie das Problem reproduzieren und leider ist auch keiner bereit das ganze wenigstens mal so zu testen, dass ich herausfinden kann wo das Problem liegt. Dementsprechend sind mir die Hände ziemlich gebunden. Ich bin mir als äußerer Betrachter relativ sicher, dass das Problem bei einer der Kernfunktionalitäten der UEFTW Treiber von cecekpawon liegt (evtl auslesen der EFI Partition auf der suche nach einer Treiber.plist). Wenn es dir den Aufwand wert ist, könntest du mal ein ROM ohne KernnextPatcher und DarBoot flashen und ich glaube du hättest keine Probleme. Andererseits würdest du auch nicht macOS booten können... Dementsprechend kann der Zeitpunkt, zu dem die UEFTW Treiber aktiviert werden, ebenfalls eine Rolle spielen, evtl also die Position im ROM (weshalb ich die Treiber im Paket nummeriert habe).

Sollte es sich durch isolierte Tests herausfinden lassen, dass zB der KernnextPatcher das Problem ist, sollte man sich bei cecekpawon melden, damit er den Fehler finden kann. Leider kann ich das Problem weder reproduzieren, noch aktuell irgendwelche Arbeit am Computer machen...

Beitrag von „derHackfan“ vom 17. Januar 2019, 13:11

[Zitat von kuckkuck](#)

Wenn es dir den Aufwand wert ist, könntest du mal ein ROM ohne KernnextPatcher und DarBoot flashen und ich glaube du hättest keine Probleme. Andererseits würdest du auch nicht macOS booten können...

Welchen Sinn macht das wenn ich kein macOS booten kann?

Was ich machen kann ist mit DBounce Step by Step (nach deiner Nummerierung) die Treiber einzeln laden und schauen wie sich das Mainboard verhält, ab dem Zeitpunkt wo das UEFI einfriert ist der schuldige gefunden, denke ich...

Deswegen schrieb ich gestern dass ich nur noch EnhancedFat tauschen/einsetzen werde, DBounce hat meiner Meinung nach nichts im rom zu suchen, das wäre wieder eine (mögliche)

Fehlerquelle mehr.

Btw.: Ich nutze rEFInd und kann ohne DarBoot macOS Mojave booten.

Werde die Fortschritte hier berichten.

Beitrag von „kuckkuck“ vom 17. Januar 2019, 13:18

[Zitat von derHackfan](#)

Welchen Sinn macht das wenn ich kein macOS booten kann?

Ausschließlich um zu testen, ob die beiden entfernten Treiber Schulc sind 😊 Wenn du KernextPatcher manuell nachlädst, kannst du auch wieder macOS booten...

Das mit DBounce und rEFInd finde ich eine tolle Sache, ich werde gerne lesen was du so herausfindet und evtl ab und zu mal einen Kommentar da lassen 😊

Beitrag von „derHackfan“ vom 18. Januar 2019, 08:20

Update:

Den FileSystem Treiber gegen EnhancedFat getauscht.

Mit DBounce die **Ozmosis High Sierra ready** Files geladen.

Ein High Sierra 10.13.4 lässt sich als APFS wie gewohnt zügig booten.

Nächstes:

Die Treiber von .ffs nach .efi extrahieren/umwandeln.

Mit DBounce die Ozmosis Mojave ready Files (min Paket) laden.

Ein High Sierra 10.13.4 oder Mojave 10.14.2 versuchen zu booten.

Beitrag von „derHackfan“ vom 19. Januar 2019, 18:59

[kuckkuck](#) kannst du mir bitte das min und das mid Paket als .efi Driver zur Verfügung stellen, mein Versuch die Treiber zu laden funktioniert nicht, ich möchte gerne den Kreuztest machen?

Update: Jetzt kann ich zwar mit DBounce das min Paket bzw. Teile (HFSPPlus.efi, Apfs.efi, Ozmosis.efi, KernnextPatcher.efi) davon laden ohne dass das UEFI einfriert, aber dafür kann ich (gar) kein OS X oder macOS mehr booten.

Irgendwie drehe ich mich da gerade im Kreis, auch weiss ich nicht ob es an der extrahierten Defaults.plist liegt oder an meiner...?

Beitrag von „kuckkuck“ vom 21. Januar 2019, 15:14

EFIs erstellst du, indem du den Body des Pe32 Images aus dem ROM extrahierst und nach .efi umbenennst 😊

<https://github.com/cecekpawon/...-Driver-Module-Extraction>

Was Ozmosis angeht, würde ich folgendes empfehlen:

"In this case, I would like to advise you to put Oz path on last from DriversList (sequenced list) instead of DriversPath, and leave LoaderPath empty by default. So DBounce will load all required drivers before finally load Oz as Chainloader."

Jedes der UEFTW Tools hat und braucht eine config/driver.plist. Wenn diese nicht vorhanden ist, kann ich mir vorstellen, dass es zu Problemen kommt... Die plist kann von verschiedenen Orten geladen werden: <https://github.com/cecepawon/UEFTW/wiki/Howto-Driver>

Wenn alles nötige durch DBounce geladen wird und jeder Treiber (auch Ozmosis) seine Plist besitzt und erhält, kannst du mal versuchen macOS zu booten. 😊 Wenn es nicht funktioniert, wäre interessant wo es hakt...

Beitrag von „derHackfan“ vom 21. Januar 2019, 17:24

[kuckkuck](#) bei allem Respekt, erzähl mir bitte nicht wie es geht, das weis ich alles und sonst wäre ich auch nicht so weit gekommen. 😊

Kannst du bitte einfach nur deine FFS aus einem ROM extrahieren um zu vergleichen?

Beitrag von „kuckkuck“ vom 22. Januar 2019, 03:19

Tut mir leid, du, ich wollte lediglich ein paar Anregungen geben, es gibt sehr viele Faktoren die eine Rolle spielen und jeder vergisst mal was... Dass man Ozmosis als Driver und nicht als Loader injected ist keine Selbstverständlichkeit und dass die UEFTW Treiber im Gegensatz zu Ozmosis keine FallBack Plist in den Treiber implementiert haben und deswegen ohne externe Plist nicht funktionieren können, sieht man auch nur, wenn man sich den Treiber mal im Disassemble anschaut. Ich dachte das wäre vielleicht interessant...

Um dir die EFIs (die ich benutze) zur Verfügung zu stellen, würde ich das gleiche wie du tun, das letzte ROM, das ich gepostet habe öffnen und einfach die Treiber per UEFITool extrahieren. Leider kann ich das aktuell nicht machen, bin für längere Zeit unterwegs und nicht in Reichweite eines Computers, tut mir leid. Bei einigen Treibern kann man die EFIs auch direkt von der Github Source laden, damit ist man auf der sicheren Seite (AppleSupportPkg, HfsPlus, PartitionDxe)

Beitrag von „derHackfan“ vom 22. Januar 2019, 06:42

Edit: EFIs und FFSs im Release: [macOS Mojave Ozmosis BETA – OZ167X-XMAX](#)