

Erledigt

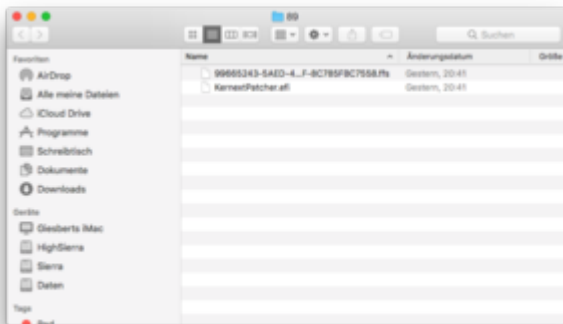
On The Fly Kernel und Kextpatcher für Ozmosis

Beitrag von „griven“ vom 14. Juli 2017, 00:06

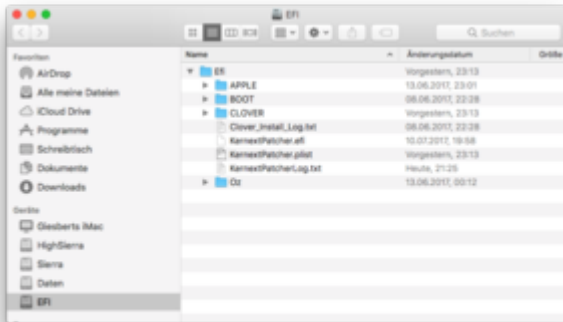
Einer der größten Kritikpunkte an Ozmosis ist die Tatsache das es mit Bordmitteln von Ozmosis bisher nicht möglich ist Kernel und Kextpatches ähnlich wie bei Clover on the Fly zu realisieren sprich wo immer Patches nötig sind (Whitelist Patch für Broadcom WLAN Karten, NVME SSD usw...) war man mit Ozmosis bisher außen vor denn die Patches ließen sich nicht auf eine ähnlich komfortable Weise wie mit Clover ins System einschleusen sondern man musste entweder selbst Hand anlegen und einen HEX Editor bemühen oder aber eine Litanei von Perl Befehlen ausführen beides keine wirklich zufrieden stellende Lösung da hierbei die Dateien dauerhaft verändert werden und man das Prozedere nach jedem Update wiederholen durfte bis jetzt...

[@cecepawon](#) hat sich diesem Problem angenommen und Abhilfe in Form eines Kernel and Kextpatchers geschaffen der sich sowohl als Driver über die UEFI Shell (nicht ganz unwichtig wenn wenig Platz im ROM ist) als auch als FFS direkt im ROM unterbringen lässt. Das Ganze greift das ExitBootServices Event ab und klemmt sich lax gesprochen zwischen OZ und den Start des Kernel durch OS-X und nimmt ähnlich wie Clover die Patches dann im RAM vor (innerhalb des prelinked Kernels) der Vorteil liegt auf der Hand da diese Methode nicht destruktiv ist sprich es werden keine Files auf der Platte verändert aber wie funktioniert das nun und was muss getan werden um es zu nutzen?

Als erstes braucht man logischerweise erstmal den Patcher selbst den man sich am besten auf der [dazu gehörenden GitHub Seite](#) besorgt (<https://github.com/cecepawon/...ter/Module/KernextPatcher>). Einmal geladen befindet sich im Downloads Ordner ein Verzeichnis mit folgendem Inhalt



Je nachdem wie man den Patcher einsetzen möchte benötigt man aus dem Verzeichnis entweder die .efi (als Treiber über die EFI Shell geladen) oder die .ffs (direkt in den ROM gebaut) Datei. Ich empfehle aktuell den Treiber noch über die EFI Shell zu laden denn er befindet sich aktuell noch in der Entwicklung und durch das laden über die EFI Shell lassen sich die Versionen ggf. schnell austauschen ohne dabei jedes mal den ROM neu flashen zu müssen. Damit der Treiber über die EFI Shell geladen werden kann muss er auf die EFI Partition im Ordner EFI abgelegt werden



zusätzlich zum Patcher selbst braucht es natürlich auch noch das File in dem die einzelnen Patches definiert werden müssen. Dieses File (KernextPatcher.plist) ist eine einfache .plist die sich an der Struktur der Kext2Patch Einträge aus der config.plist von Clover orientiert (besonders praktisch für Clover Umsteiger 😄) hier mal ein Beispiel wie die Datei aussehen könnte:

XML

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
   "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3. <plist version="1.0">
4. <dict>
5. <key>KernextPatches</key>
6. <dict>
7. <key>KernelToPatch</key>
8. <array>
9. <dict>
10. <key>Comment</key>
11. <string>Reboot fix for xcpm</string>
12. <key>Disabled</key>
13. <false/>
14. <key>Find</key>
15. <data>
16. VUIj5UFXQVZBVUFUU1BBidZBifdliftFhf8PhA==
```

```
17. </data>
18. <key>MatchOS</key>
19. <string>10.12</string>
20. <key>Replace</key>
21. <data>
22. w0ij5UFXQVZBVUFUU1BBidZBifdliftFhf8PhA==
23. </data>
24. </dict>
25. <dict>
26. <key>Comment</key>
27. <string>xcpm bootstrap</string>
28. <key>Disabled</key>
29. <false/>
30. <key>Find</key>
31. <data>
32. g8PEg/si
33. </data>
34. <key>MatchOS</key>
35. <string>10.12</string>
36. <key>Replace</key>
37. <data>
38. g8PGg/si
39. </data>
40. </dict>
41. <dict>
42. <key>Comment</key>
43. <string>xcpm bootstrap</string>
44. <key>Disabled</key>
45. <false/>
46. <key>Find</key>
47. <data>
48. idgExDwidyl=
49. </data>
50. <key>MatchOS</key>
51. <string>10.13</string>
52. <key>Replace</key>
53. <data>
54. idgExjwidyl=
55. </data>
56. </dict>
57. <dict>
```

```

58. <key>Comment</key>
59. <string>Reboot fix for xcpm</string>
60. <key>Disabled</key>
61. <false/>
62. <key>Find</key>
63. <data>
64. vgsAAABd6QgAAAAPH4QAAAAAFVlieVBVw==
65. </data>
66. <key>MatchOS</key>
67. <string>10.13</string>
68. <key>Replace</key>
69. <data>
70. vgsAAABd6QgAAAAPH4QAAAAAMNlieVBVw==
71. </data>
72. </dict>
73. </array>
74. <key>KextsToPatch</key>
75. <array/>
76. </dict>
77. <key>Preferences</key>
78. <dict>
79. <key>Debug</key>
80. <false/>
81. <key>Off</key>
82. <false/>
83. <key>SaveLogToFile</key>
84. <true/>
85. </dict>
86. </dict>
87. </plist>

```

Alles anzeigen

Ihr seht der Aufbau entspricht der Kext2Patch bzw. Kernel2Patch Sektion der config.plist von Clover es lassen sich sogar die Einträge aus der config.plist von Clover 1:1 in dieses Konstrukt übernehmen. Um das ganze nu zu nutzen muss der Patcher dem UEFI nun noch vertraut gemacht werden sofern Ihr nicht das .ffs direkt in den ROM gebaut habt sondern wie empfohlen den Weg über die EFI Shell geht kommt ihr mit den folgenden Befehlen ans Ziel.

1. EFI Shell starten -> Bios Bootmenu -> Buildin EFI Shell.
2. Filesystem mittels fs0: auf die EFI Partition der ersten Platte wechseln.

3. Mittels `cd EFI` ins EFI Verzeichnis wechseln.
4. Mit dem Befehl `bcfg driver add 0 KernextPatcher.efi "KernextPatcher"` den Treiber hinzufügen.
5. Exit eingeben und neu starten

Wenn Ihr alles richtig gemacht habt verrichtet der Patcher ab jetzt seinen Dienst und protokolliert sein Tun in einem Logfile auf der EFI Partition. Sobald Ihr alle Patches die Ihr benötigt in die `KernextPatcher.plist` eingefügt habt und alles zu Eurer Zufriedenheit läuft könnt Ihr im unteren Bereich der `KernextPatcher.plist` das Logging auch abschalten hierzu einfach im Bereich Preferences der `KernextPatcher.plist` den Wert für den Key `SaveLogToFile` auf `false` setzen.

So nun aber viel Spaß beim testen und natürlich heißen Dank an [@cecekpawon](#) für dieses geniale Ding 😄

Beitrag von „crusadegt“ vom 14. Juli 2017, 09:17

Interessante Geschichte... Vill werd ich mich auch nochmal mit Ozmosis beschäftigen 😄

Beitrag von „INTOIT“ vom 28. Oktober 2017, 12:00

Super! 😄 Hier im Forum findet man tatsächlich zu allem einen Eintrag, hier bräuchte ich aber etwas Hilfe:

Ich versuche mich gerade daran eine `Kernexpatcher.plist` zu erstellen, um den Boot graphics Fehler (10.12. HD4000) aus der Welt zu schaffen, ohne das System zu verändern.

Die Hex Werte um den `IOGraphicsFamily.kext` zu patchen sind folgende:
Find 0100007525

Replace 010000eb25

1. Wie kann ich diese Werte in der Kernexpatcher.plist integrieren, also wie komme ich auf die entsprechenden Werte oder wo kann ich diese finden?
2. Kann ich KernnextPatcher.efi und die KernnextPatcher.plist Datei auch vom Desktop aus in die gemountete Efi legen oder muss das aus dem Buildin EFI Shell gemacht werden?

Im Anhang habe ich meine KernnextPatcher.plist und ein Screenshot vom Plist Editor. Die Plist habe ich hier aus dem Beitrag entnommen, auf eine Sektion gekürzt und die Beschreibung unter Comment geändert...

Beitrag von „al6042“ vom 28. Oktober 2017, 16:54

Das sollte so aussehen...

Spoiler anzeigen

Beitrag von „INTOIT“ vom 28. Oktober 2017, 17:53



Vielen Dank! Läuft

Frage 2 hatte sich dann erledigt, nachdem ich die Anleitung nochmal gelesen hatte...
Und zur Info, funktioniert hatte es erst nach der 2. Bekanntmachung im Shell.. (bcfg driver add 0 KernnextPatcher.efi "KernnextPatcher")

Beitrag von „griven“ vom 1. November 2017, 00:48

Nice 😄😄

Beitrag von „nikoZ“ vom 17. Mai 2018, 21:00

How can I correctly add 10.13.4 PMHeart USB patch? I only need correct patch xml.
kernextpatcher.plist
Thanks.

Beitrag von „al6042“ vom 17. Mai 2018, 21:12

Hi,

put the following code into the kernextpatcher.plist:

Code

1. <dict>
2. <key>Comment</key>
3. <string>Remove 15 port limit completely in 10.13.4</string>
4. <key>Disabled</key>
5. <false/>
6. <key>Find</key>
7. <data>
8. g32UDw+DIwQAAA==
9. </data>
10. <key>InfoPlistPatch</key>
11. <false/>
12. <key>Name</key>
13. <string>com.apple.driver.usb.AppleUSBXHCI</string>
14. <key>Replace</key>
15. <data>
16. g32UD5CQkJCQkA==
17. </data>

18. </dict>

Alles anzeigen

Beitrag von „nikoZ“ vom 17. Mai 2018, 21:43

To be able to boot correctly kernexpatcher.efi and ozmosis.efi guessi have to remove ozmosis.ffs from bios right?

Thanks.

Beitrag von „al6042“ vom 17. Mai 2018, 21:51

You should put the kernexpatcher as an ffs directly into the rom itself...

Beitrag von „nikoZ“ vom 17. Mai 2018, 22:12

Got it working by editing kernexpatcher.plist according to your code then integrated kernexpatcher.ffs and ozmosis.ffs into bios. All good, all ports working correctly using latest USBInjectall and properly edited DSDT.

Thank you very much!

Beitrag von „al6042“ vom 17. Mai 2018, 22:26

Great news... 👍