Erledigt Ozmosis Patch macOS Mojave

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/clov...atform/kext_inject.c#l595
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 KernextPatcher.
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 Kernextpatcher 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?