

Ozmosis BIOS mit Ubuntu erstellen

Beitrag von „derHackfan“ vom 28. März 2018, 22:41

Wine

Ein Ozmosis BIOS mit Ubuntu erstellen, geht das überhaupt, ja das geht ...



Als erstes braucht man Unmengen von Alkohol, ich bevorzuge hier Ubuntu mit Wine und Bier auf dem Schreibtisch.



Terminal

Das Terminal lässt sich mit der Tastenkombination `Strg + Alt + T` öffnen.

Offizielle Paketquellen:

Code

1. sudo apt-get install wine

Ab Ubuntu 16.10:

Code

1. sudo apt-get install wine-stable

- In Bearbeitung -

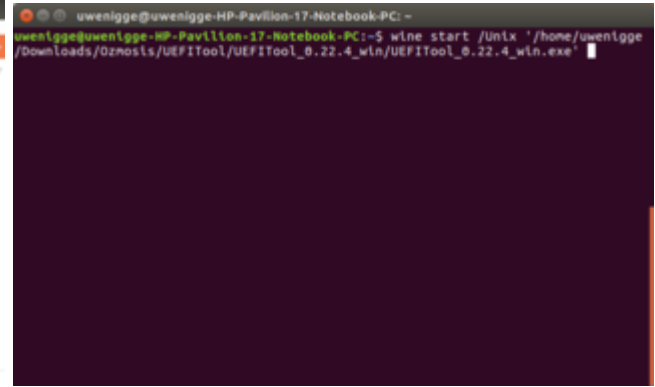
Beitrag von „derHackfan“ vom 28. März 2018, 22:42

Informationen

Weitere Informationen zum Thema UEFITool und Aptio Capsule und UEFIPatch und Ozmosis findet ihr hier -> [Ozmosis Bios für macOS High Sierra erstellen](#) und hier -> [Ozmosis BIOS individuell anpassen + Erklärung des Aufbaus](#) und hier -> [Wiki](#).

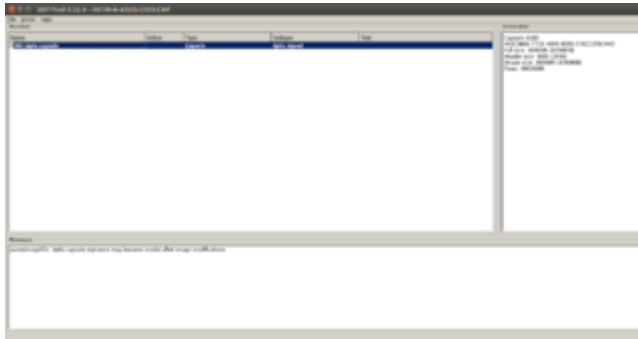
UEFITool

Fangen wir an mit dem UEFITool und einem ASUS BIOS, das rom muss da raus, aus der Aptio Capsule versteht sich und das mit "Extract Body".



Code

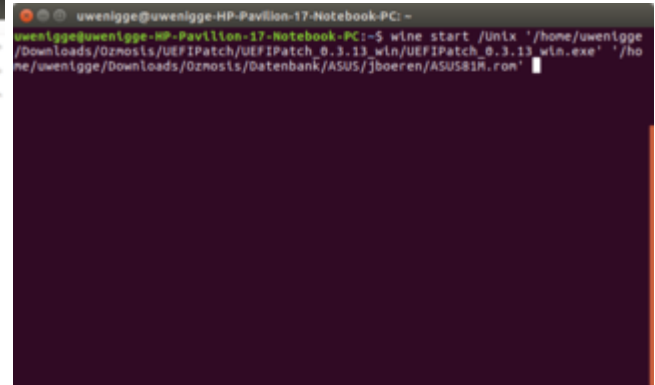
1. wine start /unix '/home/uwenigge/Downloads/Ozmosis/UEFITool/UEFITool_0.22.4_win/UEFITool_0.22.4_win.exe'



Das UEFITool mit der AMI Aptio Capsule

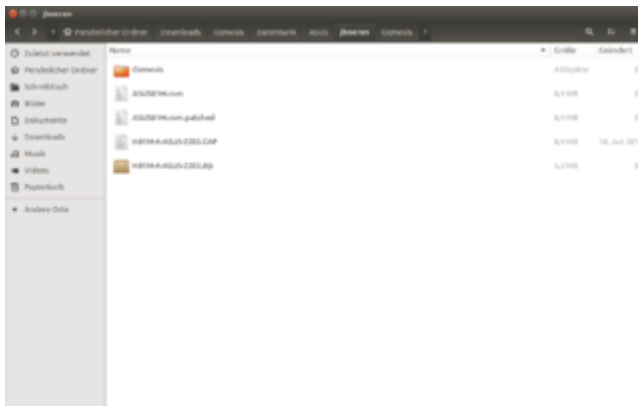
UEFIPatch

Weiter geht es mit dem UEFIPatch und demselben ASUS BIOS, wenn es was zu patchen gibt dann machen wir das auch, das Ergebnis erhält dann ein .patched ...



Code

1. `wine start /unix '/home/uwenigge/Downloads/Ozmosis/UEFIPatch/UEFIPatch_0.3.13_win/UEFIPatch_0.3.13_win.exe' '/home/uwenigge/Downloads/Ozmosis/Datenbank/ASUS/jboeren/ASUS81M.rom'`

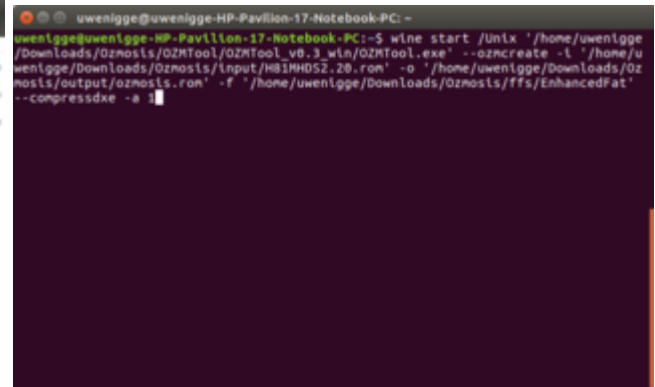
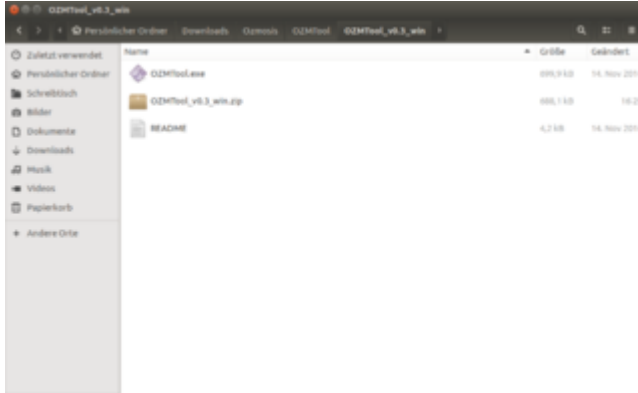


Das Ergebnis von UEFITool und UEFIPatch.

OZMTool

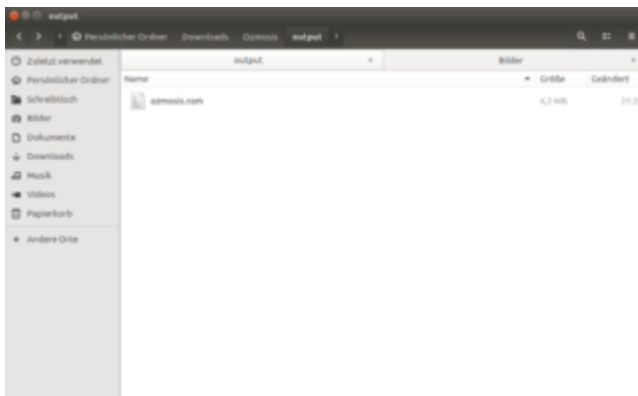
Ozmosis und seine Files werden mit dem OZMTool in das rom eingebaut, hier gibt es die Möglichkeit nur den EnhancedFat zu ersetzen oder wenn genug Platz im rom ist, alle Ozmosis

Files vollständig zu integrieren.



Code

1. wine start /Unix
'/home/uwenigge/Downloads/Ozmosis/OZMTool/OZMTool_v0.3_win/OZMTool.exe' --
ozmcreate -i '/home/uwenigge/Downloads/Ozmosis/input/H81MHDS2.20.rom' -o
'/home/uwenigge/Downloads/Ozmosis/output/ozmosis.rom' -f
'/home/uwenigge/Downloads/Ozmosis/ffs/EnhancedFat' --compressdxe -a 1



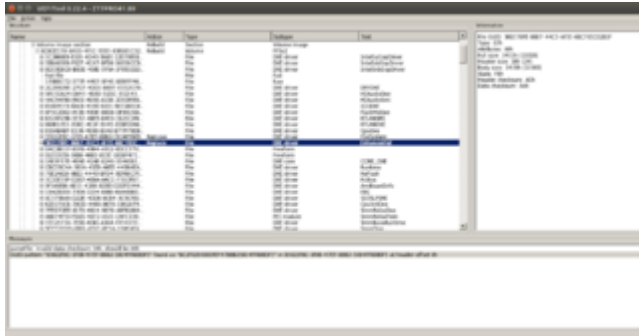
Das fertige rom im Ordner output.

- In Bearbeitung -

Beitrag von „derHackfan“ vom 28. März 2018, 22:42

EnhancedFat

Mit dem UEFITool den EnhancedFat Treiber einsetzen als Vorbereitung auf DBounce.



GUID (93022F8C-1F09-47EF-BBB2-5814FF609DF5)

Now replace the FileSystem module with EnhancedFat by right clicking on the FileSystem module and selecting Replace as is....

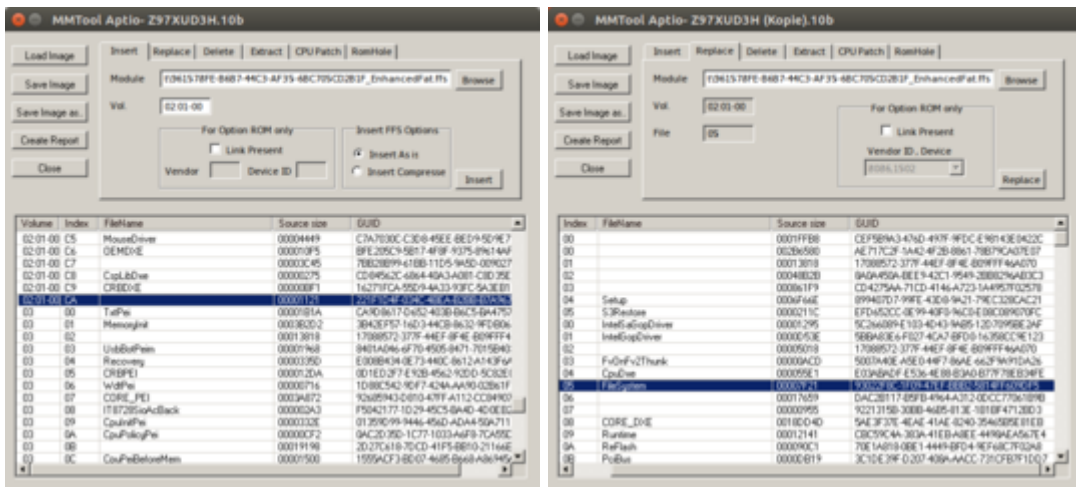
Quelle: insanelymac

Thread: [Das DBounce Experiment - APFS, HFSPPlus, Ozmosis aus der EFI laden](#)

Download: [UEFITool 0.22.4](#) | [UEFIPatch 0.3.13](#)

LAN Treiber

Die LAN Treiber mit dem OZMTool (automatisch), UEFITool oder MMTTool (händisch, manuell) löschen.



Links: Insert As is (das FileSystem vorher löschen) Rechts: Replace (das FileSystem direkt ersetzen)

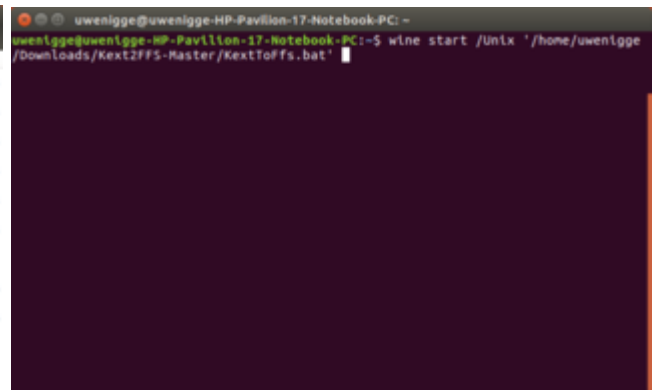
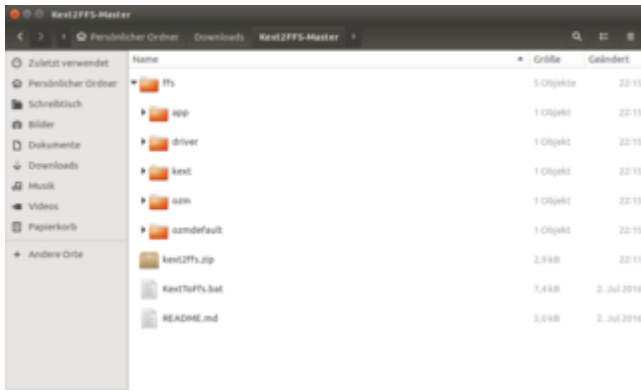
Thread: [Das DBounce Experiment - APFS, HFSPPlus, Ozmosis aus der EFI laden](#)

Download: none

- In Bearbeitung -

Beitrag von „derHackfan“ vom 30. März 2018, 21:26

KextToFFs



Code

1. `wine start /Unix '/home/uwenigge/Downloads/Kext2FFS-Master/KextToFFs.bat'`

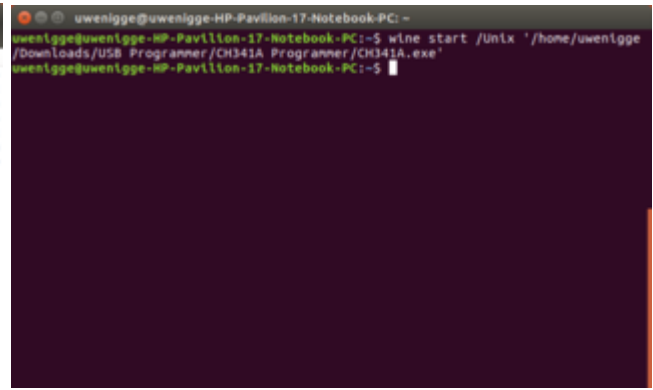
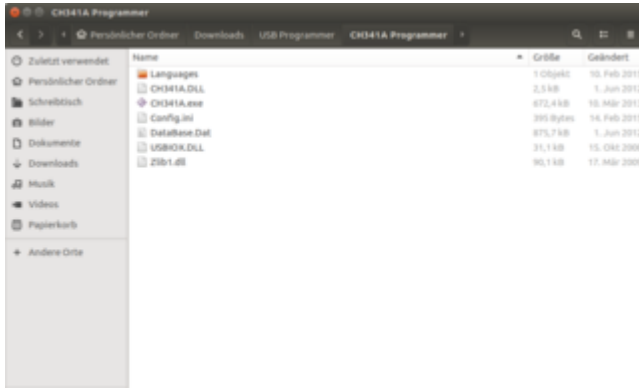
Thread: [Kext2FFS](#)

- In Bearbeitung -

Beitrag von „derHackfan“ vom 2. April 2018, 10:37

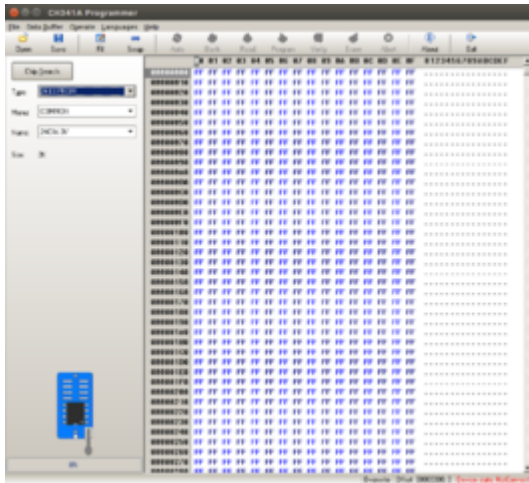
USB Programmer





Code

1. wine start /Unix '/home/uwenigge/Downloads/USB Programmierer/CH341A.exe'



Thread: [USB High Speed Programmer \(REQUEST THREAD / Write Anfragen\)](#)

- In Bearbeitung -

Beitrag von „derHackfan“ vom 2. April 2018, 17:15

Shellscript

In Ubuntu ein Shellscript ausführbar machen mit Doppelklick:

Code

1. `chmod a+x Dateiname.sh`

Code

1. `gsettings set org.gnome.nautilus.preferences executable-text-activation ask`

In Ubuntu ein Shellscript mit der Anwendung Textbearbeitung erstellen:

Code

1. `#!/bin/bash`
2. `' wine start /Unix '/home/uwenigge/Downloads/USB Programmer/CH341A Programmer/CH341A.exe'`

- Platzhalter -

- In Bearbeitung -

Beitrag von „modzilla“ vom 2. April 2018, 19:43

Im Bezug auf UEFI(Patch/Tool) kann auch die native Route genommen werden, indem man es

sich selbst kompiliert (bei mir halt das bspw. auch nur > 1 min gedauert, um die Programme zu bauen)!

Dazu muss man die folgende Befehle im Terminal ausführen:

Code

1. `sudo apt install build-essential qt4-qmake`
2. `mkdir ~/compile`
3. `curl -L -k -o ~/compile/UEFITool.tar.gz https://github.com/LongSoft/UEFITool/archive/0.22.4.tar.gz`
4. `cd ~/compile/`
5. `tar -xvf UEFITool.tar.gz`
6. `cd ~/compile/UEFITool-0.22.4`
7. `qmake UEFTool.pro`
8. `make -j9 ##(da ich 8 Prozessorkerne besitze, d.h. logische Prozessoren +1)`
9. `./UEFITool`

Gleiches gilt für UEFIPatch:

Code

1. `cd ~/compile/UEFITool-0.22.4/UEFIPatch`
2. `qmake UEFIPatch.pro`
3. `make`
4. `./UEFIPatch -(whatever)`

Alternativ könnt ihr es auch mal mit den angehängten tools probieren, die ich bereits kompiliert habe!

Dazu müsst ihr dann einfach mit dem Terminal in den Ordner gehen, an dem ihr die Dateien heruntergeladen habt und dann könnt ihr sie mit `./DATEINAME` ausführen!