

Kann man eigentlich Apple Silicon spoofen?

Beitrag von „EdD1024“ vom 28. Juli 2021, 13:37

Hallo zusammen,

Lese gerade [diesen Artikel auf heise](#) und frage mich, ob OpenCore auch Apple Silicon vorgaukeln könnte, um auf diese Features doch ran zu kommen...

Hab leider zu wenig Plan, um es selbst einzuschätzen...

Beitrag von „apfel-baum“ vom 28. Juli 2021, 13:50

ich vermute-

eher nicht, da ggf. die instructionsets nicht vorhanden sind und so ins leere gelaufen wird- so die nicht abgefangen werden- und oder die befehlssätze werden softwareseitig (treiber?) abgefangen -umgesetzt, dann "vielleicht"

lg 😊

Beitrag von „EdD1024“ vom 28. Juli 2021, 13:54

Naja, es gibt ja diverse Kexte, die Mac-only Instructions abfangen. Einen 64-Kerner juckt es wenig, ob die Neural Engine bestimmte Aufgaben erledigt oder ein paar der gerade schlafenden Kerne...

Aber klar, gutes Argument...

Beitrag von „griven“ vom 28. Juli 2021, 14:21

Vor allem müssten diese Features dann aber auch in den X86 Binaries enthalten sein und ich denke das unabhängig davon, ob es gelänge Apple Silicon zu spoofen hier der Hase im Pfeffer liegt.

Beitrag von „EdD1024“ vom 28. Juli 2021, 15:43

Ich glaube der Code ist da, nur in einer "if-Abfrage", ansonsten müsste Apple eine Menge Dateien doppelt liefern, oder?

Beitrag von „griven“ vom 28. Juli 2021, 16:04

Was Apple ja auch macht (Stichwort universal Binary) denn die die M1 Chips verstehen (zumindest von sich aus) den X-86 Code nicht und umgekehrt die Intel Maschinen schon gar nicht den für die ARM basierten Apple Silicon Chips. Also so einfach ist das Ganze dann halt doch nicht immerhin sprechen wir hier von zwei vollkommen unterschiedlichen Chip Architekturen mit komplett unterschiedlichen Befehlssätzen.

Beitrag von „EdD1024“ vom 28. Juli 2021, 16:16

Sicher alles richtig, aber was ist schon einfach. OC, WEG, Lilu oder FakeSMC zu schreiben war bestimmt auch nicht einfach. Ist es aber unmöglich?

Vielleicht gibt es bald eine UnlockM1Features.kext 😊

Beitrag von „DSM2“ vom 28. Juli 2021, 16:26

So ziemlich unwahrscheinlich... (UnlockM1Features.kext) Schau dir mal den Code an...