

AMD Radeon 6650 XT, 6950 XT und 6900 XTXH ohne GPU-ID Spoofing

Beitrag von „hObelware“ vom 26. Februar 2023, 13:28

Hallo allerseits,

ich stelle hier mal meine Kernel Extension für die eigentlich nicht unterstützten Ausbaustufen von Navi21 (RX 6950 XT und RX 6900 XTXH) und Navi23 (RX 6650 XT) zur Verfügung.

Ein Vorwort

Warum weg vom DeviceID Spoofing? ..

DeviceID Spoofing hat meines Erachtens nach "unsaubere" und sogar "dreckige" Lösungsqualitäten und sollte eigentlich nur als temporärer Workaround in Betracht gezogen werden.

unsauber, weil das GPU-ID Spoofing offensichtlich nicht bis zu letzten (untersten) Ebene greift. Wer mit gespoofter GPU-ID mal im IORegistryExplorer im GFX0 Device den "compatible" Key ansieht, wird dort nur die originale und nicht die gespoofter ID vorfinden. .. Bisher hatte ich zwar noch nie das Problem, daß das relevant wird, .. richtig konsistent ist dieser Zustand aber auch nicht.

dreckig, weil hier Device-unkritisch in die Geräte-/Treiberinstanzierung eingegriffen wird, und mit der Festlegung der DeviceID nach PCI-Pfad alle Geräte in diesem Steckplatz die gleiche DeviceID (und damit den gleichen Teiber) aufgedrückt bekommen, .. ganz egal, was da drin steckt. Die Flexibilität der Hardwarekonfiguration ist damit (zumindest für die Grafikkarte) dahin.

Spätestens wenn man mal - z.B. im Notfall - ne andere Grafikkarte verbauen muß, holt einen das vielleicht wieder ein .. mit Sicherheit aber, wenn das Tauschgerät ne komplett andere Architektur aufweist.

Außerdem finde ich DeviceProperties per config.plist oder SSDT/DSM nur für Geräte angemessen, die man nicht austauschen kann, weil fest aufm Board verbaut (Ethernet, Onboard Audio, etc.) Bei allem, was tauschbar ist, sollte eine dynamische Lösung angestrebt werden ..

Das Ziel

Ein PersonalityProvider, der wie jeder normale Treiber gerätesensitiv die Treiberinstanzen erzeugt, und alle zusätzlichen DeviceProperties on-demand-only als All-In-One Lösung vergibt.

Das Ergebnis

Ursprünglich hatte ich das Projekt nur für die RX 6650 XT umgesetzt, da das Konzept im Grunde aber für jedes Device funktioniert, dass über ID Spoofing läuft (nicht nur Grafik), habe ich diese Kext mal auf alle mir bekannten, zu spoofenden Navi20 erweitert.

Die IOPersonality zu erzeugen ist ja nicht sonderlich kompliziert und bei GitHub habe ich am Wochenende noch eine Lösung gefunden, die DeviceProperties via IOService zu übergeben (vgl. [hier](#)). Ich nutze eine leicht abgewandelte Version dieser Codes, der Credit hierfür gebührt in jedem Falle Voight-Kampf. .. Danke

Installation/ggf. Anpassungen

Die Kernel Extension muß wie jede andere auch in Kernel->Add eingebunden werden.



Mindestvoraussetzung ist Navi23 Support, also macOS 12.1 ..

Die Bridge SSDT(SSDT-BRG0.aml oder vergleichbares) ist nicht mehr nötig (zumindest hier unter macOS 13.2.1 .. gerne testen und hier Bescheid geben ..) und alle DeviceProperties für die Navi20 .. bei mir: `PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x0)/Pci(0x0,0x0)/Pci(0x0,0x0)` .. müssen auch weg (wir wollen ja explizit davon weg 😊).

Unmodifiziert gibt die Kext nur die Properties `model` (Graka-Name), `ATY,FamilyName` und `ATY,DeviceName` (die letzteren für `MetalDeviceName`) weiter. Ich habe bei `model` übrigens "AMD" weggelassen, damit bei "über diesen Mac" kein Zweizeiler draus wird, wer will kann das je wieder hinzufügen.

Wenn noch `FramebuffersEdits`, `PowerPlayTables` oder ggf. weitere Kosmetika etc. mitgegeben werden sollen, können diese in der `info.plist` unter `IOKitpersonalities->entsprechendes Root Device->IOProviderMergeProperties` als Kind-Elemente eingetragen werden.

Achtung!: Hierbei müssen die erwarteten Datentypen passen (meist `Data`, .. vermute ich) .. am besten vorher mit gespoofter ID mit `IORegistryExplorer` das Device `GFX0` abgleichen. Strings sind z.B. Null-terminated in Mac Roman Encoding zu übergeben (vgl. [hier](#))

Mein Plan ist das Ganze nochmal als .dext (DriverExtension) anzusetzen, so wie Apple das vorsieht. Leider weiss ich noch nicht genau, wann ich Zeit dafür finde und wie das mit den Developerlizenzen dafür aussehen muß, damit das auch überall geht (und nicht nur lokal bei mir).

Beitrag von „kaneske“ vom 27. Februar 2023, 18:48

Schicke Sache, anpinnen [griven](#) ?

Beitrag von „griven“ vom 28. Februar 2023, 08:16

Aber auf alle Fälle 😊

Der Pin ist gepinned...

Beitrag von „jan2000“ vom 28. Februar 2023, 12:49

Sehr geiles Projekt, danke fürs Teilen.

Sehe ich das richtig, dass kein PCI Pfad angegeben werden (muss)? Falls ich nun mehr als eine Grafikkarte im System habe, wie kann ich z.B. unterschiedliche PPTs übergeben? Eine rein theoretische Frage für mein Verständnis.

Beitrag von „griven“ vom 28. Februar 2023, 13:03

So rein vom drauf und drüber gucken würde ich sagen über den Parameter IONameMatch denn es werden ja nicht beide GPU's den gleichen IOName haben sondern sich unterscheiden zum

Beispiel als GFX0 und GFX1...

Beitrag von „hObelware“ vom 28. Februar 2023, 14:47

genau, über IONameMatch, .. entweder ne Kopie des Root Devices mit anderem IONameMatch angeben (für unterschiedliche DeviceProperties) oder aus IONameMatch ein Array machen und für jeden Namenstreffer einen (string) Eintrag für resultierend gleiche Properties.

Im Grunde kannst Du IOClass ACPIProviderMergeProperties (hier die jeweiligen RootDevices) dazu verwenden, an jedes beliebige IOPCIDevice oder IOUSBDevice DeviceProperties zu übergeben, von dem Du den IOName und/oder die DeviceID (IOPCIMatch) kennst, ganz unabhängig vom PCI Pfad .. das läuft vom X6000 Treiber völlig losgelöst. Nur die übrigen IOPersonalities (nicht RootDevices) machen die Treiberanbindung.

Im Detail arbeitet Voight-Kampfs IOClass so, dass sie beim Testen ob die Klasse für das Device instanziiert bleiben soll (probe), in der Probe() Routine statt zu Testen, die DeviceProperties schubst und dann immer NULL meldet (für: kann weg). Danach wird die Instanz wieder abgebaut.

Beitrag von „SchmockLord“ vom 28. Februar 2023, 17:35

Sehr cool, Danke @[hObelware](#),

Werd ich gleich mal ausprobieren, hab ne 6900 XTXH.

Update: Super, hat funktioniert. Endlich wird meine 6900 XTXH auch sauber als solche ausgewiesen. Vorher stand überall nur Navi21.

Valid	Possibly
✓ GFX0	
GPU Name	Navi 21 [Radeon RX 6900 XT]
GPU Device ID	0x73AF1002
Quartz Extreme (QE/Ci)	Yes
Metal Supported	Yes
Metal Device Name	AMD Radeon RX 6900 XTXH
Metal Default Device	Yes
Metal Low Power	No
Metal Headless	No

Aber die Geekbench Scores sind unverändert:

ID	Model	Compute API	Platform	Compute Score
102749	MacPro7,1 [AMD Radeon RX 6900 XT]	Metal	macOS 14.0.0	238500
102743	MacPro7,1 [AMD Radeon RX 6900 XT] Compute Engine	OpenCL	macOS 14.0.0	119694
102711	MacPro7,1 [AMD Radeon RX 6900 XT] Compute Engine	OpenCL	macOS 14.0.0	120168
102626	MacPro7,1 [AMD Radeon RX 6900 XT]	Metal	macOS 14.0.0	241364

Beitrag von „hObelware“ vom 28. Februar 2023, 19:13

[SchmockLord](#) der Treiber ist ja auch der gleiche wie vorher .. die Taktung VRAM/GPU wird sowieso vom VBIOS bestimmt.

Unterm Strich ist der Betriebszustand exakt der gleiche, wie vorher, .. nur der Weg dahin ist dynamisch und der DeviceTree bildet die Wirklichkeit ab ..

Beitrag von „kaneske“ vom 28. Februar 2023, 19:35

[hObelware](#) ganz tolle Sache, eine Sorge weniger für so einige. Bin begeistert weil ich es auch so Nutzer freundlich ist.



Beitrag von „Sascha_77“ vom 1. März 2023, 06:52

Da wird der Kext Updater wohl bald ein Update erfahren. 😊

Beitrag von „sisxfx“ vom 24. März 2023, 03:18

Hallo. Zunächst einmal Entschuldigung, wenn ich etwas falsch schreibe, da ich Google Translate verwende. Derzeit besitze ich einen 6650 XT, der mit der üblichen Methode arbeitet (Geräte-ID als FF730000 + GPU-SPOOF.aml mit Boot-Argumenten agdpmod=pikera) und gut funktioniert. Aber ich habe gelesen, dass einige Leute diese Methode verwenden und besser arbeiten.

In meinem Fall verursachte es einen Neustart und ich sah den Anmeldebildschirm nicht.

Die Schritte, die ich befolgt habe, sind:

- 1- Behalten Sie alle grünen Kext- und Boot-Argumente bei
- 2-Entfernen Sie aus den Geräteeigenschaften die lange PCI (diejenige, die sich auf die AMD dGPU bezieht)
- 3-Fügen Sie die kext AMDRadeonNavi2xExt.kext.zip in den nächsten Ordner ein und laden Sie sie in die config.plist

Irgendeine Idee, womit ich Probleme bekomme?

Danke

Beitrag von „ozw00d“ vom 24. März 2023, 07:28

[sisxfx](#) if you speak English it's not necessary to translate it. Simply write in English. Most of all here understand it.

Beitrag von „sisxfx“ vom 24. März 2023, 12:22

Danke [ozw00d](#)

Hello. I own an XFX 6650 XT which works with the usual method (device id as FF730000 + GPU-SPOOF.aml + with boot arguments agdpmmod=pikera) and works fine. But I read that some people use this method and was trying to implement it because seems a better way.

I am using last version of Opencore 0.90

In my case, it caused a reboot and I didn't see the login screen.

The steps I followed are:

1- Keep whatevergreen.kext and boot arguments (If I understand well, that part doesn't need to be removed)

2-From the device properties, remove the long PCI (the PciRoot(0x0)/Pci(0x1,0x0)/Pci(0x0,0x0)/Pci(0x0,0x0)/Pci(0x0,0x0))

3-Put the kext AMDRadeonNavi2xExt.kext in the kext folder and load it to the config.plist

Any idea why isn't working? I can upload the config.plist or something else if that helps

Thanks

Beitrag von „hObelware“ vom 26. März 2023, 18:43

hi there, .. yes, I would be best to upload an anonymized config for further investigation .. probably add an ioreg save with active ID Spoofing too

Beitrag von „sisxfx“ vom 27. März 2023, 00:27

Hi,

Thanks for the reply.

Here is the config.plist file [config.plist](#)

According to the opencore log the kext is being loaded:

19:355 00:023 OC: Prelinked injection AMDRadeonNavi2xExt.kext (AMDRadeonNavi2xExt.kext)
- Success

19:375 00:019 OC: Prelinked injection AMDRadeonNavi2xExt.kext v1.0.1

So currently, I am using two aml files and is working fine(ACPI folder):

SSDT-BRG0.aml ---> Attached DSL [SSDT-BRG0.dsl](#)

SSDT-6650XT-73EF-GPU-SPOOF.aml ---> Attached DSL [SSDT-6650XT-73EF-GPU-SPOOF.dsl](#)

I have a ImacRegistry.ioreg file (5mb compressed) with the active spoofing here --> [ImacRegistry.zip](#)

Beitrag von „SchmockLord“ vom 28. März 2023, 10:04

@[hObelware](#) Ich hab deine Kext-Methode ja schon länger im Einsatz. Soweit hat sie funktioniert. Aber gestern beim Update auf 13.3 nicht mehr. Habs bestimmt 10 Mal probiert. Weder Update noch als Full Installer. Er hat dann irgendwann auch sinngemäßg "GFX0 missing" angezeigt im Verbose Mode beim Booten.

Hab dann nur auf die alte Variante gewechselt, SSDT-BRG0 und die device-properties und er ist direkt durchgebootet. Sowohl der Updater als auch Full Installer.

Wollte dich das nur wissen lassen.

Beitrag von „hObelware“ vom 29. März 2023, 23:22

meine Kiste hat gestern auch lange überlegt, bis es weiter ging beim Update (aber es ging schlussendlich weiter) .. aber mit 13.3 is der VDA Decoder tot, mit Spoofing gehts aber alles .. ich geh der Sache nach .. Danke für die Info

Beitrag von „utopia“ vom 5. April 2023, 14:23

First of all, Thank you so much for your work
I have Rx 6650 XT and your kext worked like a charm on the first try,
I'm using Ventura 13.3, and I have no issues at all except my OPENCL geekbench 6 score is around 20000 which seems to be very low.
did I mess something up?

Beitrag von „hObelware“ vom 6. April 2023, 15:43

I usually don't care about benchmark scores at all. Additionally (from what I read) they may vary from OS Version to OS Version ..

Are the scores better when using spoofing? I didn't test that, but I believe there shouldn't be a big difference, since my kext utilizes the very same drivers as spoofing does.

For the time being, I use spoofing, since I ran into VDADecoder issues after upgrading to 13.3 .. my next step will be, to implement the rx 6650 support as driver extension .dext (as intended by apple), maybe that will solve DRM Decoding .. but I'm rather short on spare time right now, so I can't predict, when exactly I will be able to do something about that.

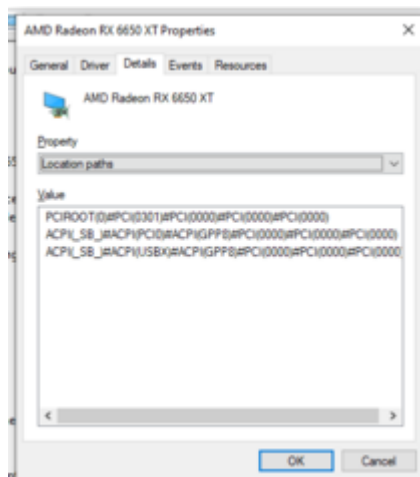
Beitrag von „utopia“ vom 6. April 2023, 19:46

I never tried it myself but some others got way higher scores than me (x3 times more)

I'm also on 13.3 and VDADecoder is not working as well

I've tried to spoof the card before but i always fail I'm not sure why, the device-id seems to never change, and location paths on Windows report 2 ACPI paths, i only took the first one as _SB_PCI0_GPP8, and it shorter than usual or is this normal?

i tested multiples booting the efi from the USB and didn't work, (didn't want to risk your working kext on my disk efi)



Beitrag von „hObelware“ vom 7. April 2023, 16:07

[utopia](#) it seems, there is something not properly set regarding USBX (please upload your ACPI folder from OC)

besides that .. your PCI path to set DeviceProperties must be

PciRoot(0x0)/Pci(0x3,0x1)/Pci(0x0,0x0)/Pci(0x0,0x0)/Pci(0x0,0x0)

the (for spoofing mandatory) SSDT-BRG0 has to be edited too, but for that I'd need a IORegistryExplorer save from your running system (kext based)

Beitrag von „utopia“ vom 7. April 2023, 16:17

I just fixed it :D, I had two unnamed bridges in my path, it wasn't showing on Windows but from IORegistryExplorer I could see its full path:

IOService:/AppleACPIPlatformExpert/PCI0@0/AppleACPIPCI/GPP8@3,1/1OPP/pci-bridge@0/1OPP/pci-bridge@0/IOPP/GFX0@0

from this, I was able to edit the SSDT to match the path (credit to @etorix)

Also: I made a new geek bench test on the GPU with the spoofing and I got the exact same result as your KEXT, both were very low on OpenCL

<https://browser.geekbench.com/v6/compute/298161>

Beitrag von „hObelware“ vom 7. April 2023, 16:31

from your windows screenshot I read, that USBX (unusually) shares the PCI Path with the

you should review the USBX declaration (SSDT-USBX?)

Beitrag von „utopia“ vom 7. April 2023, 16:36

EDIT:

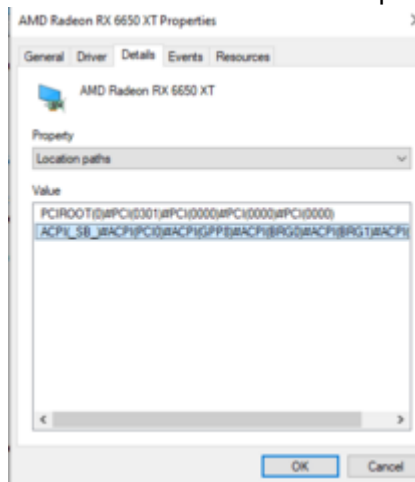
Based on your suggestion, I recreated a new USBX SSDT using SSDTIME and it managed to remove the USBX paths from Windows, but the OpenCL scores are still the same

EDIT2:

I managed to fix the issue by enabling Shaneee's patch in (Kernel/patches):

https://github.com/AMD-OSX/AMD_Vanilla

the new scores match the specs: <https://browser.geekbench.com/v6/compute/298388>



Beitrag von „startergo“ vom 26. April 2023, 04:21

Zitat von hObelware

Unfortunately, DRM Decoder does not work with this solution.

Could it be because the `AMDRadeonX6000_AMD RadeonHWServicesNavi` is not injected properly?

Also `AMDRadeonX6000_AMDNav21GraphicsAccelerator` is missing from IO Registry at all. I am attaching an IO Registry backup for comparison. The `IOProbeScore` is set to 6502, which is higher than the original by only 1, whereas the `IOProbeScore` of the `AMDRadeonX6000_AmdGpuWrangler`, which injects correctly is 2 units higher than the original?

Beitrag von „GenuineJakob“ vom 11. Juli 2023, 14:32

Hi [hObelware](#),

if you don't have the time to investigate this any further, could you please upload the Xcode project or the relevant parts of the code to GitHub? Maybe the Whatevergreen devs can look into it and fix DRM.

Still many thanks for the kext that is available so far!

Beitrag von „startergo“ vom 11. Juli 2023, 15:13

[Zitat von GenuineJakob](#)

Hi [hObelware](#),

if you don't have the time to investigate this any further, could you please upload the Xcode project or the relevant parts of the code to GitHub? Maybe the Whatevergreen devs can look into it and fix DRM.

Still many thanks for the kext that is available so far!

The Xcode project this is based on is on GitHub. It is actually Apple Original open source code. There is a link in the first post.

Alles anzeigen

Beitrag von „fabi99“ vom 16. August 2023, 13:18

Hallo @h0belware,

Vielen Dank für deine praktische kext. Unter Ventura wird meine rx6900 xtxh ohne Probleme erkannt und die Grafikbeschleunigung funktioniert. Unter Sonoma ist dies leider nicht mehr der Fall. Mit spoofing bekomme ich ein Bild, mit der kext allerdings nicht, es kommen Meldungen wie error = 3 no such process und es wird nie in den Grafikmodus gewechselt. Planst du die kext für Sonoma an zu passen?

Viele Grüße

Beitrag von „h0belware“ vom 17. August 2023, 09:18

nein, bislang liegt das Projekt auf Eis, da ich Sonoma vermutlich nicht mehr betreiben werde und für DRM auch schon unter Ventura keine praktikable Lösung in Sicht ist, außer ich ließe die kext von Apple selbst signieren (was sicher nicht passieren wird) .. sorry

Wer sich daran weiter versuchen will, kann dem github Link zu voight kampf in Post 1 folgen. Ich habe im Grunde genau diese Funktionalität benutzt, um die DeviceProperties zu schubsen (.. ich hab zusätzlich noch versucht die Strings der Properties als OSDData zu casten, damit man „Klartext“ weiterreichen kann .. leider erfolglos .. OSDynamicCast greift nicht und OSRequiredCast führt zu schweren Abstürzen beim Booten).

Die eigentliche „Leistung“ der kext ist die Zuweisung der diversen Treiberklassen an die Geräte, das läuft alles in der info.plist ab und ist eigentlich ein simpler Injektor. (Wer das lesen kann, wird es auch als solches erkennen.)

Für DRM sieht der Forschungsstand meinerseits so aus, dass der DRM VDADecoder keine unsignierten Caller akzeptiert. Ne Möglichkeit wäre das Konzept ggf. als *.dext statt als *.kext aufzusetzen (mal im Apple DriverKit stöbern), da greifen möglicherweise die Apple Signaturen der originalen Treiberklassen, dann haperts aber vermutlich mit der Initialisierung durch

OpenCore.

knock yourselves out .., zum Weiterbasteln reicht die info.plist, mein Programmcode entspricht dem oben genannten, meiner heißt erstmal nur anders, (meine Erweiterungen sind in meiner angehängten kext aus Stabilitätsgründen bislang nicht „scharf“ geschaltet)

Beitrag von „Hutfred“ vom 4. September 2023, 14:33

Servus,

gibt es so eine Lösung auch für die RX6700XT?

Grüße

Beitrag von „hObelware“ vom 4. September 2023, 14:59

nein .. Navi22 wird vom Apple Treiber (bislang) nicht unterstützt.

.. dazu wird es wohl auch nicht mehr kommen, da es auf der Basis keine "Pro"- (Workstation-) Karten gibt.

Beitrag von „startergo“ vom 13. September 2023, 16:31

Zitat von hObelware

I have the kext signed by Apple itself (which will certainly not happen) .. sorry

Wait so you say If I notarize the kext the DRM will work?

Beitrag von „hObelware“ vom 15. September 2023, 09:25

nope .. it is not that easy. you have to port the kext to a dext (driver extension), which has to be signed.

and then MAYBE DRM will work .. but thats a capital maybe (as writren)

Beitrag von „kollaesch“ vom 14. Januar 2024, 13:48

[hObelware](#)

hi kollaesch hier aus DD. Das ist ja Jahre her. (und dass Du jetzt auf Mac gelandet bist ...)

Bin gerade voll geflashed, dass Du mir mein aktuelles Problem gelöst hast.

Weiss nicht, wie man hier Direktnachrichten schreiben kann. (Profil durfte ich wegen Berechtigungen nicht aufrufen.) Schreib mich mal an.

Beitrag von „hObelware“ vom 20. Januar 2024, 17:59

na dorr [kollaesch](#), .. ist ja wirklich Jahre her! (meine Telefonnummer ist aber seit 20 Jahren die gleiche)



Mac/Hackintosh betreibe ich als Haupt-/Produktivsystem seit 2011, die Landung ist daher auch schon irgendwo im Erinnerungsnebel verschollen .egal.

Wenn Du Dein Problem mit meiner Kext lösen (naja) konntest: fein!, eine echte Lösung (auch DRM) ist das hier: <https://github.com/ChefKissInc/NootRX>

Das ist die Treibererweiterung als Lilu PlugIn. Der Einsatz dieser Kext erfolgt, wie bei meiner, ohne DeviceProperties und GFX0 Bridge. Optimal fährst Du damit nen iMacPro1,1 mit Lilu.kext und ohne WhateverGreen.kext - und damit auch ohne agdpmod=pikera. Damit geht dann grafikseitig alles.

Falls Du auf QuickSync (für Sidecar z.B.) angewiesen bist, und nen iMac als SMBIOS bevorzugt wird, muß WhateverGreen für die IGPU (mit agdpmod=pikera) **nach** NootRX in der config geladen werden, sonst gibts ne KernelPanic. Da geht auch alles, wie es soll. Ob mit der Kombi Stabilitätsprobleme auftreten (glaube ich nicht), habe ich nicht mehr genügend lange getestet, da ich mir in der BlackWeek ne RX 6800 günstig geschossen habe. Die RX 6650 XT hab ich wieder verhökert.

Falls Du den Code nicht selber kompilieren willst, findest Du die Builds auch bei GitHub unter Actions->CI als Artefacts.zip

lg

Beitrag von „startergo“ vom 18. Februar 2024, 15:00

[Zitat von hObelware](#)

na dorr [kollaesch](#), .. ist ja wirklich Jahre her! (meine Telefonnummer ist aber seit 20 Jahren die gleiche)



Mac/Hackintosh betreibe ich als Haupt-/Produktivsystem seit 2011, die Landung ist daher auch schon irgendwo im Erinnerungsnebel verschollen .egal.

Wenn Du Dein Problem mit meiner Kext lösen (naja) konntest: fein!, eine echte Lösung (auch DRM) ist das hier: <https://github.com/ChefKissInc/NootRX>

Das ist die Treibererweiterung als Lilu PlugIn. Der Einsatz dieser Kext erfolgt, wie bei meiner, ohne DeviceProperties und GFX0 Bridge. Optimal fährst Du damit nen iMacPro1,1 mit Lilu.kext und ohne WhateverGreen.kext - und damit auch ohne agdpmod=pikera. Damit geht dann grafikseitig alles.

Falls Du auf QuickSync (für Sidecar z.B.) angewiesen bist, und nen iMac als SMBIOS bevorzugt wird, muß WhateverGreen für die IGPU (mit agdpmod=pikera) **nach** NootRX in der config geladen werden, sonst gibts ne KernelPanic. Da geht auch alles, wie es soll. Ob mit der Kombi Stabilitätsprobleme auftreten (glaube ich nicht), habe ich nicht mehr genügend lange getestet, da ich mir in der BlackWeek ne RX 6800 günstig geschossen habe. Die RX 6650 XT hab ich wieder verhökert.

Falls Du den Code nicht selber kompilieren willst, findest Du die Builds auch bei GitHub unter Actions->CI als Artefacts.zip

lg

Alles anzeigen

Thanks. You may want to update your link to point to <https://github.com/ChefKissInc/NootRX>. At the moment it just opens non existent <http://nootrx/>

Beitrag von „al6042“ vom 18. Februar 2024, 16:17

Done... thanks for the heads up... 🐞

Beitrag von „Scanduz“ vom 3. April 2024, 14:54

Code

1. Hi everyone, my name is Angelo, and I'm making you a new member. I followed this guide to activate graphic effects on my rx 6650 xt asus rog strix installed on a mac pro 5.1 with mac os monterey. I got the apple loading and then black screen flashing. I was kindly asking for help in activating the graphic effects on the card
-

Beitrag von „Scanduz“ vom 21. April 2024, 15:39

Code

1. Thanks for your cooperation, I solved the problem myself
-

Beitrag von „mitchde“ vom 21. April 2024, 20:33

Hi, normaly ist should work using the kext <https://github.com/ChefKissInc/NootRX>.

It may be **your** problem if you use an **orig. MacPro** and **not** an Hackintosh!

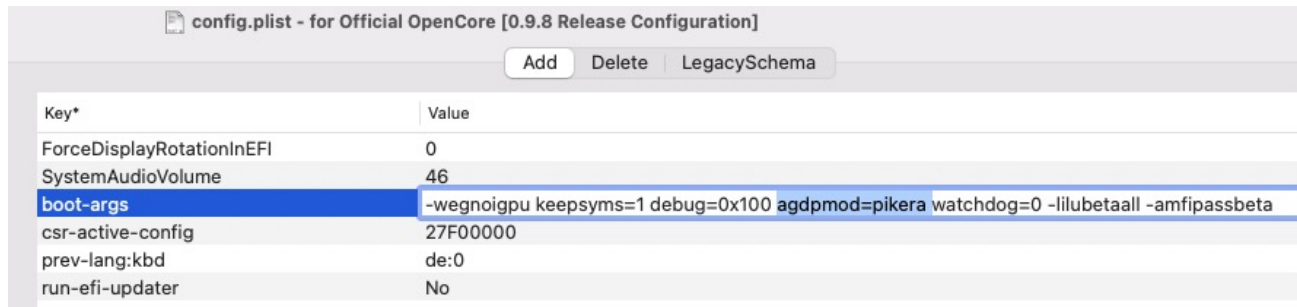
Try to ask the developer ChefKlssInc about that.

Beitrag von „Scanduz“ vom 21. April 2024, 22:29

thank you very much, I will try to hear from the developer

Beitrag von „mitchde“ vom 22. April 2024, 09:10

Beside the kext, you **must** have (in OpenCore config.plist) the **agdpmod=pikera** boot argument.



If you come from non AMD gpu card this boot argument may be missed. Than also black screen.

last but not least - use not Whatevergreen kext together with the NOOTRX.kext!!!

Only one of them - and as an last try use both but load NOOTRX.kext **before** WEG kext. Before means NOOTRX.Kext is before WEG.kext in the OC config.plist(If you use OCLP for orig. macs that also uses OC config.plist) .

Beitrag von „hObelware“ vom 22. April 2024, 11:29

[mitchde](#) .. agdpmod=pikera will simply have no effect at all, if WhateverGreen.kext is not used (because this is implemented there) ..

NootRX.kext implements the same modification to AppleGraphicsDisplayPolicy as WhateverGreen.kext does, but only when iMacPro1,1 SMBIOS is used. if you want to run that, you should need nothing but NootRX.kext.

if you want to run another SMBIOS (e.g. some iMac, which uses IGPU and dGPU, or MacMini) then make sure NootRX.kext is loaded prior to WhateverGreen.kext in config.plist and everthing should work as intended. you run into a dependency problem, as NootRX.kext will not work with several values modified by WhateverGreen, which results in a kernel panic if you load these kexts the other way around. when using WhateverGreen.kext agdpmod=pikera is mandatory as well, as their initiaization process reverts the modification NootRX.kext did to AppleGraphicsDisplayPolicy.

Beitrag von „Scanduz“ vom 22. April 2024, 13:23

so if I understand correctly I have to load nootrx.kext first and then WhateverGreen.kext and insert agdpmod=pikera

Beitrag von „hObelware“ vom 22. April 2024, 15:28

yes, that would be the correct way .. but nootrx not first! only prior to whatevergreen .. lilu has to be first at any circumstance

Beitrag von „Scanduz“ vom 22. April 2024, 17:58

1) install nootrx.kext

2) install WhateverGreen.kext and boot-arg **agdpmod=pikera**

the Mac in question is an Apple Mac Pro 5.1 and I use OCLP

Beitrag von „hObelware“ vom 22. April 2024, 18:22

i got no experience on booting real macs via OLCP .. lilu.kext has to be present to load nootrx and whatevergreen as they are lilu plugins

Beitrag von „Scanduz“ vom 22. April 2024, 19:01

Code

1. I tried loading nootrx first and then WhateverGreen.kext but the screen flashes before entering
-

Beitrag von „hObelware“ vom 23. April 2024, 08:05

before entering what? the screen flashes once when loading the driver .. that is normal

Beitrag von „Scanduz“ vom 23. April 2024, 12:15

Code

1. no unfortunately it flashes continuously with a black screen
-

Beitrag von „hObelware“ vom 23. April 2024, 22:03

[Scanduz](#) maybe you could upload your EFI (as a zip) .. I might take a look 😊

Beitrag von „Scanduz“ vom 23. April 2024, 22:45

Code

1. thanks so much, I have attached my current efi that I use to boot Sonoma on a test hdd

Beitrag von „Scanduz“ vom 25. April 2024, 11:17

Code

1. I'm doing a test with Mac OS Ventura, let's see what happens. I kindly ask you to analyze the updated efi that I have published. Thank you for your support and patience

Beitrag von „fabiosun“ vom 25. April 2024, 18:51

[Scanduz](#) as i said you on an italian forum in a real Mac Pro 5.1 due cpu insdtruction limitation is not possible to have a 6600/6650 on OSX different from Monterey...

I thought i was clear with you in this thing 😊

Beitrag von „Scanduz“ vom 25. April 2024, 22:05

[fabiosun](#) I'm stubborn but after several trials I finally understood that it's not possible to go beyond Monterey. I thought I would be able to share a guide, but this time I have to abandon the work I was doing. 😊