

Zum aktuellen Stand: iMessages im Hackintosh geht! (aber nicht überall)

Beitrag von „MacGrummel“ vom 17. Juni 2015, 08:13

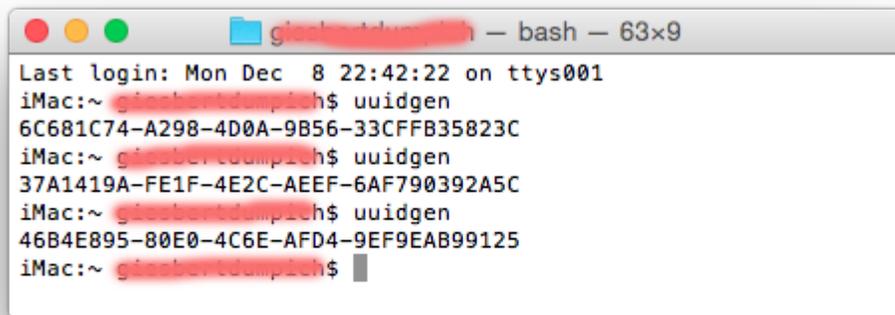


Was braucht man für iMessages?

Immer:

Das Protokoll-Programm [iMessage Debug2](#) zum Auslesen der Daten am Eigenbau oder auch an einem Original-Mac.

Einen Hackintosh mit selbst erstellten einzigartigen [Serien- und UUID-Nummern](#) , zudem solltet Ihr Euch noch einen zufällige PlattformUUID vergeben um sicherzustellen, das diese einmalig ist. Hierzu einfach im Terminal den Befehl `uuidgen` eingeben und das mehrfach wiederholen (nur um sicher zu gehen).



```
Last login: Mon Dec 8 22:42:22 on ttys001
iMac:~ g...h$ uuidgen
6C681C74-A298-4D0A-9B56-33CFFB35823C
iMac:~ g...h$ uuidgen
37A1419A-FE1F-4E2C-AEEF-6AF790392A5C
iMac:~ g...h$ uuidgen
46B4E895-80E0-4C6E-AFD4-9EF9EAB99125
iMac:~ g...h$
```

Und was geht nicht?

Zur Zeit gibt es keine sinnvolle Lösung, einen Rechner mit älterem Ozmosis, also 894m oder noch älter, und einen Rechner mit Tonies komplettem Bastelsatz, also Chimera zum Laufen zu bringen. Allerdings kann man bei 90% der Chimera-Rechner auch problemlos Chameleon installieren. Mit etwas Mehraufwand geht es bei den gleichen Rechnern mit Clover aber deutlich besser. Und eine Chameleon/Chimera-Version für El Capitan gibt es bisher ja auch nicht..

Bei OZMOSIS 1479: nur noch die Adresse im NVRam, wo die am Mac ausgelesenen Werte eingetragen werden.

Spoiler anzeigen

Bei Clover: Den Clover Configurator, um die Werte zu berechnen und in die Clover config.plist einzutragen.

Spoiler anzeigen

Bei Chameleon: Brauchen wir komplett generierte System-Zahlen und die Chameleon-Version 2.3svn und das [NVRam-Paket](#)

Spoiler anzeigen

Die zur Zeit schnellste und einfachste Möglichkeit,

iMessages auf den Hackintosh zu bringen, ist, das Eintragen dieser Werte, ob selbst generiert oder von einem Original-Mac in eine Clover - config.plist mit dem dazu gehörenden Configurator. Wenn weiter so viele Leute jetzt auf Clover setzen, sind die Bugs da sicher auch bald raus.

Wir wollen ja nichts klauen, wir wollen nur unsere Software normal nutzen. Es gibt zur Zeit 44 verschiedene Online-Dienste von Apple, bei den meisten gibt es irgendeine Art von Identifikation. Also Vorsicht, Leute! Lieber die Nummern zwei, drei Mal vor der ersten Benutzung überprüfen, als gesperrt zu werden!

Ist alles erledigt startet Ihr Euren Rechner nun neu meldet Euch aber bitte nach dem Reboot nicht direkt an iMessage oder FaceTime an sondern überprüft erstmal, ob die Werte auch übernommen wurden. Am einfachsten geht das mit dem Tool [iMessageDebug2](#), da stehen dann alle wichtigen Werte drin.

Werkzeuge

[iMessageDebug2](#)

[Clover Configurator](#)

[NVRam für Chameleon](#)

[Apple MAC Hack](#)

Zusätzliche Links

[Der komplette Thread zu diesem Thema](#)

[SmBIOS](#)

[Beschreibung Clover Configurator](#)

Das erste Werkzeug zur kompletten ID-Zahlen-Generierung, [Apple MAC Hack](#), ist noch in frühem Alpha-Stadium.

[iMessage, Griven's Tread 1 und die ganze Unterhaltung](#)

Kaum ein Thema ist in unserem und anderen Hackintosh-Foren in den letzten Wochen und Monaten so heftig diskutiert worden wie der Versuch, den Apple-Dienst iMessage/Nachrichten auf unsere Rechner zu bringen. Und die Stufen, in denen Apple das weitgehend verhindert (hatte).



Nachdem der Dienst über Jahre mehr lustlos und ohne besondere Beachtung sowohl unter Macs als auch auf Hackintosh-Rechnern eben einfach so mit lief (oder auch nicht lief..), ist er mit der Einführung von OS X Yosemite und iOS 8 zu einer gemeinsamen SMS & Mitteilungszentrale für Rechner, iPads und iPhones geworden und wurde mit einem Mal wirklich interessant. Relativ fix kamen wir dahinter, dass wir jetzt außer den bekannten Werten für iTunes, der Netzwerk-Adresse von en0 und der Seriennummer, auch andere Hardware-Zahlen brauchten, im Apple-System MLB (Main Logic Board) und ROM genannt.

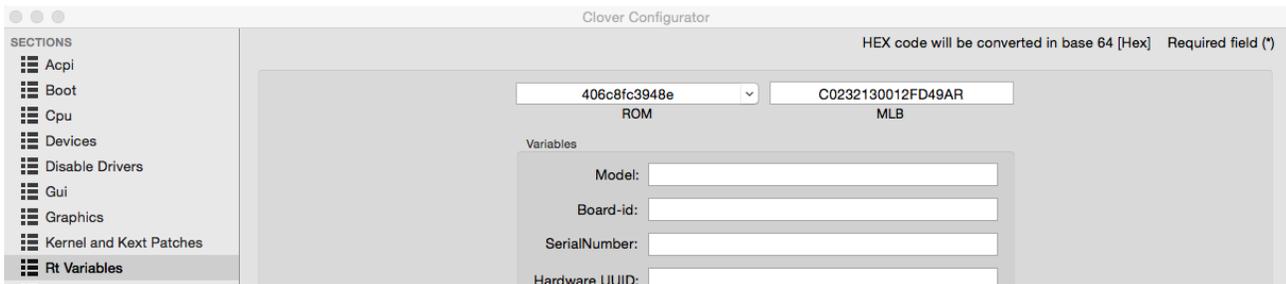
Als MLB-Wert nimmt Apple die Serien-Nummer des Motherboards. Die findet man bei Apple-Rechnern dann auch irgendwo dort auf dem Board zusammen mit dem Herstellungsort.

Der ROM-Wert entspricht zu einem großen Teil der Netzwerk-Adresse des ersten Firewire-Anschlusses fw0, von den dort 16 Zahlen und Buchstaben werden einfach die vier mittleren weg gelassen. Bei älteren Mac-Rechnern steht diese Nummer zusammen mit der Rechner-Serien-Nummer auf einem kleinen Aufkleber auf dem Gehäuse auf der Rück-, Innenklappen- oder Unterseite.

Diese Werte können bei normalen fest konfigurierten Rechnern direkt zur eindeutigen Identifikation genutzt werden, sie waren aber bisher für Hackintosh-Rechner ohne Bedeutung. Auch jede Netzwerk- oder WLAN-Karte, egal ob fest ins Board integriert oder als PCI-E-Karte, hat solche eindeutigen und zugewiesenen MAC-Adressen.

Nur hat leider FakeSMC an all unsere Rechner die gleiche Board-Serial verteilt (zu bewundern in den System-Informationen der MacPro-Typen als Seriennummer (Prozessormodul): C021...), wir haben sie ja auch nicht wirklich gebraucht.

Mit Yosemite (bei den älteren Systemen ab Mitte Dezember auch..) reichten dann diese generierten Adressen unserer Selbstbau-Rechner nicht mehr aus, wir brauch(t)en diese beiden Nummern von Original-Macs, einfach ausgelesen mit [iMessageDebug2](#).



In Clover mit dem Configurator kein großes Problem: die Nummern eines Macs in die Felder ROM und MLB auf der Seite der Rt-Variablen eingegeben und los gehts.

Auch beim OZMOSIS-BIOS brauchte man zu den Zahlen nur die richtige Adresse auf dem NV-Ram zu kennen und eintragen. Aber da kam schon der erste Haken: OZMOSIS braucht eine 17-stellige MLB-Nummer, die meisten Mac-Rechner der Generation vor den Intel-i-Prozessoren haben aber nur eine 13-stellige Board-Nummer, die OZMOSIS selbstständig auf 17 Stellen auffüllt. Da aber nur gültige Nummern-Paare den Zutritt erlaubten, war hier für viele Rechner dann Schluss.

Die Nutzer von Chameleon und Chimera scheiterten dann spätestens an der dritten notwendigen Nummer, die noch an Apple gesendet werden muss, die System-ID.

Die kann auch ohne Probleme aus der Hardware-UUID berechnet werden. Das erledigt problemlos der Clover Configurator, die hat eigentlich jeder Rechner mit 64-Bit-Mac-System. Selbst, wenn es kein Problem ist, diese zu generieren, sie wird bei der iMessage-Abfrage aus dem NV-Ram an Apple übertragen und dort stand sie bisher halt bei Chameleon nicht, deshalb waren hier die Chameleon-Rechner raus und es kam mit ganz wenigen im Netz verbreiteten Nummern-Paaren zu einem kurzfristigen Clover-Boom.

Bis kurz vor Weihnachten war es nun möglich, mit einem Paar aus ROM und MLB beliebig viele Rechner mit ihren eigenen System-IDs bei Apple anzumelden.

Das wurde dann auf den einen Wert des Rechners reduziert, der zur richtigen Zeit gerade im Netz angemeldet war und noch ein paar Tage später dann auf den System-ID-Wert zurück gesetzt, der als erster angemeldet gewesen ist.

Wie geht es nun weiter?

Woher die ROM-Werte kommen, ist ja schon lange klar und die stehen ja auch auf den älteren Rechnern, wie oben beschrieben. Die 13-Stelligen MLBs sind inzwischen weitgehend entziffert: 2xHerstellungsort, 1xJahr, 2xWoche, 4xProdNr, 4x exakteModel-ID, alle bezogen auf das Motherboard.

HackBook

F4:37:B7:A8:38:2E
ROM / MAC Address

88FB09FF-6E33-4C1D-B659-60EFF1BE2987
Hardware UUID

5A0026F6-C276-4D89-8D77-F437B7A8382E
System UUID

CK035CB066D **W80230ZG88GCA**
System Serial Logic Board Serial

MacBook Pro (13-inch, Mid 2009)

Apple UPPER
 Whatever lower

Generate

Bei den gerade auch für modernere Basisrechner notwendigen 17-stelligen Werten fehlt leider noch ein großer Teil, die Datenbasis ist einfach zu klein, es fehlen einfach noch ausreichend Werte von echten Macs.

ROM und MLB bilden sozusagen das Tor, durch das der Rechner mit seiner als UUID generierten System-ID zum i-Portal eintreten kann, bis Dezember beliebig viele IDs, jetzt geht nur noch eine durch. Da auch in der MLB die Hardware genau beschrieben ist, sind hier weitere Verknüpfungen durch Apple denkbar, aber natürlich aufwendig.

Da auch sehr viele echte Macs hier durch die Prüfung gefallen sind, vergibt Apple seit März einen Token für jeden User als zweite Authentifikation, in dem Festplatte, Rechner und UUID eingebaut werden. Es ist so zwar deutlich einfacher geworden, sich bei Apple in den Diensten zu registrieren, aber dafür kann man nicht mehr mal eben mit der Festplatte aus einem anderen Rechner ins iMessage-Netz.

Oft bekommt man trotz komplett korrekter Zahlen die Mitteilung, dass man den Apple-Support kontaktieren soll.

Entweder ist da aus irgendeinem Grund das Erstellen des Tokens schief gelaufen. Wenn die Zahlen im iMessageDebug stimmen, braucht man sich vor dieser viertel Stunde auch nicht fürchten. Sind ja Zahlen wie an einem echten Mac..

Oder man steckt einfach in der Warteschleife fest, die Apple eingeführt hat, um bestehende Apple-IDs zusätzlich zu schützen. Dann kommt je nach Support-Nachfrage nach 3-10Tagen die Meldung auf das iPhone und den Rechner, dass ihre Apple-ID jetzt für den und den Rechner neu benutzt wird und dann läuft alles.

Zum Verkürzen dieser Wartezeit kann man nun entweder das Passwort der bestehenden Apple-ID einfach mal ändern oder gleich für den neuen Hackintosh eine neue Apple-ID erstellen.

Leider hängt das nicht an uns. Im Augenblick ist es gerade mal wieder etwas leichter, sich bei iMessage/FaceTime anzumelden, aber letztlich sind wir da halt direkt von Apple abhängig, wie groß bei denen die Zahl und die Tiefe der Bearbeitungen ist..

Dank geht für die Arbeit an iMessage an Meklort & SJ_Underwater für das neue modifizierte FileNVRAM, jaymonkey, holyfield, Griven, MacGrummel und die breite Hackintosh-Community für viel Test-und Programmier-Arbeit und die Entwickler von iMessageDebug und an die Entwickler von Apple Mac Hack

MacGrummel

Beitrag von „hObelware“ vom 17. Juni 2015, 12:45

Danke für Deine Zusammenfassung, allerdings würde ich gern anmerken, daß

Code

1. `sudo nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:platform-uuid=6AD165EF-FAE3-4725-A5BE-C3CD74323068`

bei mir unter Ozmosis 1479 keine Auswirkungen auf die SystemID hat. Bei Ozmosis 894m ging

das noch wunderbar.

Um unter Ozmosis 1479 die SystemID zu setzen, muß

Code

1. `sudo nvram 4D1FDA02-38C7-4A6A-9CC6-4BCCA8B30102:HardwareSignature=6AD165EF-FAE3-4725-A5BE-C3CD74323068`

verwendet werden. Die korrespondierende Hardware-UUID wird dann auch korrekt generiert.

LG

Beitrag von „MacGrummel“ vom 17. Juni 2015, 21:53

Da hast Du recht, das sollte da auch nicht stehen! Ich weiß garnicht, wie das da hin gekommen ist, danke für den Hinweis!! Wird sofort geändert!

Beitrag von „Griven“ vom 17. Juni 2015, 23:07

Beitrag ist unter <http://hackintosh-forum.de/ind...ntryID=327#profileContent> auch in der WIKI zu finden 😊

Beitrag von „QSchneider“ vom 25. Juli 2015, 22:52

[MacGrummel](#)

Bei meinen aktuellen Recherchen zu Clover bin ich vielleicht auf die Antwort auf deinen Satz in deiner Anleitung gestoßen.

Zitat

SmUUID und Custom-UUID müssen aus irgend einem Grund hier aber vertauscht eingetragen werden

Google bitte einmal "Clover v.2 Instructions". Im post #23 "iCloud fix" ist der Grund für die Vertauschung aus meiner Sicht erklärt.

Wenn ich es richtig lese, wird aus der Custom-UUID die Hardware UUID errechnet und muss nicht mehr explizit angegeben werden.

Dies deckt sich mit meiner aktuellen Konfiguration bei der nur ROM, MLB und Custom-UUID (incl. Inject System ID) benötigt werden, die Hardware UUID bzw. SmUUID dagegen leer ist, sie aber trotzdem in der Hardwareübersicht und iMessage Debug auftaucht.

Dies wäre dann insofern interessant, da dann eine neue Systemdefinition nicht zu Problemen mit iMessage führen dürfte.

In meinem konkreten Fall hatte ich vorher gar nichts im Bereich SMBIOS angegeben. Clover hat für mich daher die Systemdefinition Macmini6,2 beim Booten festgelegt. Nachdem ich eine externe GPU eingebaut hatte wurde daraus plötzlich ein iMac13,1. Da ich aber verhindern wollte, dass mir Clover meine Systemdefinition mit jeder Hardwareänderung verändert, habe ich im SMBIOS explizit iMac13,1 ausgewählt. Dabei wurde natürlich keine SmUUID angelegt, trotzdem blieb die Hardware UUID gleich.

Spricht also dafür, das es sich verhält wie im posting von slice erwähnt ... oder liege ich falsch ?

Beitrag von „MacGrummel“ vom 26. Juli 2015, 15:00

[QSchneider](#): Dass die beiden UUIDs mathematisch direkt zusammenhängen stand ebenfalls im Post 1: es reicht aus, einen Custom-UUID-Wert einzutragen, wenn der Rechner nicht beide automatisch nach Wunsch ausgibt. In meinem Clover-Rechner ist dieser Part auch völlig leer. Seitdem es nicht mehr nötig ist, die Werte eines originalen Macs zu übernehmen. Wenn ich den Clover-Rechner als meinen Mini starten lassen möchte, muss ich das natürlich ändern.

Woher die nicht in der PListe stehenden Werte kommen, hab ich da auch beschrieben: die

stehen in der FakeSMC. Wenn man vergisst, die Serien-Nummern für Board und Rechner zu ändern, kommen die halt auch von da. Und sind bei jedem gleich! Genau deshalb war Deine Idee richtig, eine feste Definition vor zu geben. Die UUIDs werden vom Board vorgegeben, die anderen Werte kommen aus der FakeSMC oder eben der config.plist.

Mein Quo läuft deshalb auch problemlos mit der Serial seines Gehäuses, der Syst-Def als Powermac G4 und der Board-ID als MacPro.. Die UUIDs wurden bei jedem Ozmosis-BIOS-Update leider erneuert, sodass ich sie dann doch auch in den NVRAM eintragen musste

Beitrag von „QSchneider“ vom 26. Juli 2015, 16:39

Hallo MacGrummel, danke für die umfassende Antwort, mein Posting war auch nicht als Kritik wegen der Unvollständigkeit deiner Anleitung gedacht.

Der Aufhänger war lediglich, das ich über deinen, von mir zitiertem Satz gestoßen bin und vorher auf der englischsprachigen Seite den Post gelesen hatte.

Mein Beitrag bezieht sich konkret nur auf den Spoilerblock "Bei Clover ..." und dreht sich nur um die Clover-, bzw Clover Configurator spezifischen Einträge

Der bisherige Satz, "das aus unbekanntem Gründen irgend etwas vertauscht werden muss", verwirrt mMn mehr als er in einem "Kochrezept" hilft.

Vielleicht macht es den Spoiler deutlicher, wenn dort stehen würde, was man nun genau im Clover Configurator eintragen muss und dies sind nach meinem Stand

- ROM, MLB (in RT Variables)
- und Custom-UUID + Inject System ID (in System Parameters)

Der Eintrag SmUUID aus dem Bereich SMBIOS des CloverConfigurator (welcher die Hardware UUID darstellt) muss NICHT gefüllt werden.

Wäre das so korrekt ausgedrückt ?

Beitrag von „MacGrummel“ vom 26. Juli 2015, 17:15

Das Problem mit dem Verdrehen stammt aus der Zeit, als es notwendig war, die komplette Datenreihe von einem originalen Mac zu übernehmen, also Serien-Nr., Board-Serien-Nummer, Hardware- und System-ID und die ROM-Nummer, also die Mac-Adresse des Ethernet-en0-Ports. Die hat man halt über iMessageDebug am Original ausgelesen und dann VERKEHRT herum in die Clover-PListe eingetragen, damit der Clover-Rechner die gleichen Daten hat(te).

Und zumindest bei meinen Clover-Rechnern brauch(t)e ich keine UUID in die PListen oder den Colver-Configurator eintragen, weil sie vom Board und BIOS vorgegeben werden. Oder eben BEIDE..

Das steht in dem etwas älteren [Thread zum gleichen Thema](#), da war auch die ständige Veränderung der Apple-Anmeldung im letzten Winter nachzuverfolgen. Als sich nicht mehr viel verändert hat, hab ich aber halt eine aktualisierte Version der Anleitung geschrieben, um weitere Verwirrungen zu vermeiden.

Beitrag von „Wowadriver“ vom 24. August 2015, 10:25

Hi Leute,

nachdem ich heute meine vom originalen Mac Pro 2013 ausgelesenen Werte ins NVRAM geschrieben und einen

Neustart zum Übernehmen durchgeführt habe startet der hacki nicht mehr. Also das Apple Logo kommt und der

Balken zieht durch, jedoch wird der Montior dann schwarz also ob er in den Standby geht und kommt die Meldung

das kein Signal vorliegt... Liegt das möglicherweise das ich die Definition von Mac 3.1 auf MAC Pro 6.1 geändert habe?

Falls ja wie kann ich das wieder rückgängig machen...

War der Meinung es wäre besser alle Daten des Ausgelesenen Gerätes zu übernehmen. Damit es keine Probs mit

iMessage gibt. Habe das jetzt über meine 970 GTX laufen, hilft es vielleicht diese auszubauen und über Onboard zu

versuchen?

Gruß

Beitrag von „MacGrummel“ vom 24. August 2015, 10:57

Bei vielen MacPro-Installationen stört der AppleTyMCEDriver.kext. Den kannst Du einfach raus werfen, dann geht meist alles wieder. Und die Syst.-Definition sollte durchgehend EINEM Modell entsprechen. Sonst bekommst Du vielleicht keine Probleme mit dem Start, aber mit dem Anmelden bei den Apple-Diensten.

Beitrag von „Wowadriver“ vom 24. August 2015, 11:54

Hobbit: Hat mit dem -x leider nicht geholfen. Habe ihn durch starten über onboard Grafik wieder ins Leben gerufen.

~~Was mir noch aufgefallen ist, hatte das mit dem -x ins Nvram über einen Sudo Befehl geschrieben. Anschließend startete der Hack mit der GTX 970 aber mit extremen Grafikfehlern. Habe dann wieder onboard aktiviert, musste jetzt aber feststellen dass der Hack jedesmal mit -x starten, weil kein Sound usw geht. Wie kann ich das BootArgument wieder entfernen?~~

MacGrummel: ~~Wenn ich das richtig verstanden habe, dann meinst du mit durchgehend einem Modell entsprechend, dass all die Informationen die aus dem originalen Mac Pro ausgelesen wurden, auch in den Hack übertragen werden sollen?!~~

~~Fakt ist das habe ich nämlich gemacht. Wenn ich jetzt mit iMessageDebug die Werte des Hacks auslese, entsprechen die exakt allen Werten des Originals. Lediglich den ROM Wert habe ich durch MAC Adresse des Hacks geändert. Ist doch richtig so oder soll ich den Rom Wert des originalen Mac Pros ebenfalls eintragen?~~

Folgendes wurde erreicht: Es wurden alle, auch der Rom Wert, de ausgelesen originalen Mac verwendet! Fazit iMessage usw läuft von sofort! Ohne Wartezeit oder Aktivierungszeit! Problem ist, dass jetzt der Hack als Mac Pro 6,1 definiert ist. Laut der Anleitung wie man eine 970 GTX zum laufen bekommt, muss dieser jedoch als MAc3,1 deklariert werden. Somit läuft meine GTX 970 nicht. Wie Hobbit erwähnt hat, kann ich ihn mit -x in den SAfemode starten bringt mich jedoch nicht weiter, weil ich im Safemode ja nicht bleiben will.

Als brotarg wurde wie in der Anleitung beschrieben:

```
sudo nvram 7C436110-AB2A-4BBB-A880-FE41995C9F82:boot-args="kext-dev-mode=1 nvda_drv=1"
```

verwendet. Habt ihr einen Rat wie ich die Grappa wieder zum laufen bekomme? Falls ich die Diskussion lieber in einem separaten Thread starten soll, bitte sagen, jedoch ist der Grafikkarten Thread geschlossen...

P.S.: @ MacGrummel die .kext hatte ich gelöscht

Gruß

Wowadriver

Beitrag von „heimlik“ vom 24. August 2015, 15:44

Ich habe ein seltsames iMessage Problem: iMessage klappt generell gut (Clover) – außer mit einigen Kontakten, mit denen ich teilweise schon jahrelang problemlos chatte, die aber auch ganz neu sein können.

Zustand ist folgender an meinen 3 Geräten:

Apple MacPro: iMessage kein Problem mit allen.

iPhone: kein Problem mit allen

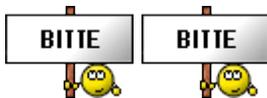
Hacky: kein Problem mit manchen, erhält alle Nachrichten wenn ich woanders schreibe, kann aber nicht aktiv an bestimmte Personen schreiben. Dann kommt sofort das rote (!).

Habe bereits an allen Geräten iMessage und iCloud komplett abgemeldet/angemeldet usw –das Problem bleibt bestehen.

Es scheint auch keine Rolle zu spielen, von wo nach wo (Telnummer, Mailadresse, iCloud-Mailadresse) ich Leuten schreibe. Es ist auch egal, ob die iPhones oder Macs am anderen Ende haben – es klappt einfach nicht "rauszutelefonieren". Hat jemand eine Idee?



Beitrag von „Wowadriver“ vom 25. August 2015, 14:34



Niemand einen Rat?

Beitrag von „Griven“ vom 25. August 2015, 21:16

Wenn Du eine MacPro 6,1 Systemdefinition fährst reicht das installieren der Webtreiber für NVIDIA Karten allein nicht aus. Der MacPro6,1 (und iMAC 15,1) unterscheiden sich in der AppleGraphicsDevicePolicy von den anderen Modellen was dazu führt das man beim verwenden dieser SMBIOSe ohne eine Anpassung der AppleGraphicsControl.kext buchstäblich in die Röhre guckt 😊 So wird es behoben:

1. AppleGraphicsControl.kext aus /S/L/E auf den Desktop kopieren
2. Rechtsklick auf AppleGraphicsControl.kext machen und Paketinhalt anzeigen auswählen
3. In das Verzeichnis /Contents/PlugIns/AppleGraphicsDevicePolicy.kext/Contents/ navigieren und die darin enthaltene info.plist zum bearbeiten öffnen
4. nach Mac-F60DEB81FF30ACF6 suchen und folgendes

Code

1. `<key>Mac-F60DEB81FF30ACF6</key>`
2. `<string>Config1</string>`

ändern zu

Code

1. `<key>Mac-F60DEB81FF30ACF6</key>`
2. `<string>none</string>`

5. Die info.plist speichern und den so geänderten Kext mit dem Kextutility erneut installieren. [Rechte reparieren](#) und Cache neu bauen nicht vergessen.
6. Reboot und Grafik genießen.

Beitrag von „kruemelnase“ vom 25. August 2015, 21:44

Gibt auch noch eine einfachere Lösung/Möglichkeit, als Bootargument

Code

1. `nv_disable=1`

eingeben, dann bootet der Rechner erstmal ohne Grafikbeschleunigung, und dann das Tool [AGPDfix](#) ausführen, das patcht den Kext automatisch. Reboot und fertig.

Beitrag von „Griven“ vom 25. August 2015, 21:58

Ist denn nv_disable=1 ein offizieller KernelFlag oder ist das Bootloader Argument? WoWaDriver setzt OZ ein und da bin ich mir ziemlich sicher, dass mit nv_disable=1 gar nichts passiert solange es sich dabei nicht um einen offiziellen Kernelflag handelt...

Beitrag von „WowDriver“ vom 26. August 2015, 19:56

[griven](#) & [kruemelnase](#) Ihr seid meine Helden des Tages! Habe beides probiert und beides hat funktioniert. Mit dem Tool sogar im Handumdrehen fertig! Einfach wie Hobbit es gepostet hat erstmal über die GTX 970 mit dem "-x" Befehl starten, sodass man im sicheren Modus das Tool ausführen kann. Neustart fertig! Jetzt habe ich einen kompletten MacPro 6,1 laufen mit all seinen originalen Daten und iMessage usw alles läuft und sogar ohne Wartezeit! Plug and Play Quasi! Und ja ich benutze die letzte Ozmosis Version...

Gruß



und vielen Dank nochmal an alle hier!

Beitrag von „Griven“ vom 2. September 2015, 20:36

Es geht auch nach wie vor mit generischen Daten sofern das Format stimmt...

Wichtig bei generischen Daten ist, dass die 2 Faktor Authentifizierung aktiviert ist und idealerweise der Apple ID auch eine Kreditkarte als Zahlungsinformation hinzugefügt wurde.