

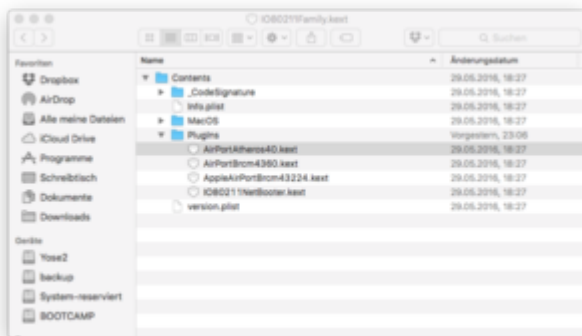
Erledigt

## Atheros AR9285 DW1520 Kext

Beitrag von „griven“ vom 6. Juni 2016, 22:53

Na gut, dann ein wenig ausführlicher erklärt (achtung es folgt Technik Foo)...

Die AR9285 VenId168c und Dev Id 002b ist eigentlich für sich genommen schon ziemlich kompatibel da nahe an dem dran was Apple in diversen Geräten auch selbst verbaut es gibt nur einen Haken die für das ansteuern des WLAN's notwendige IO80211Family.kext springt aufgrund der Device ID nicht von sich aus auf die Karte an (Apple benutzt hier eine andere Device ID). Grundsätzlich befinden sich in der IO80211Family diverse Plugins die für jeweils andere WLAN Karten Typen verantwortlich zeichnen:



Der für uns in diesem Fall richtige Einstieg ist das AirPortAtheros40.kext Plugin in ihm befinden sich die Informationen über alle unterstützten WLAN Karten von Atheros:



Unsere Karte trägt die DeviceID 2b und wenn wir mit der Liste oben vergleichen gibt es unter den kompatiblen Karten keine mit der DeviceID 2b wohl aber eine mit der DeviceID 2a und so wird die WLAN Unterstützung nicht geladen obwohl die 9285 voll kompatibel ist. Um jetzt an WLAN Unterstützung zu kommen gibt es verschiedene Wege dies zu erreichen. Am Einfachsten wäre es unsere DeviceID ebenfalls in die info.plist des Plugins einzutragen was auch ohne weiteres funktionieren würde. Diese Lösung hat aber einen entscheidenden Haken denn jedes Systemupdate würde unsere Änderungen gnadenlos überschreiben sprich der Schritt müsste also nach jedem Update wiederholt werden. Fazit eine einfach und effektive Lösung aber leider keine sehr praktikable. Ein weiterer Ansatz wäre die Verwendung eines Injectors oder eines

PCIID fakers (ToledaARPT oder FakePCIID) auch dieser Ansatz funktioniert allerdings füge ich auf diese Weise weitere Kexte dem System hinzu an die ich wieder denken muss wenn ich zum Beispiel mal dazu gezwungen bin das System neu aufzusetzen. Ein weiterer Pferdefuß bei dieser Lösung ist, dass sie uns unter Umständen nicht in der Recovery oder im Installer zur Verfügung steht sprich es wäre keine Umfassende Lösung. Der bei weitem universellste und somit auch sicherste Weg das Problem zu lösen ist es direkt auf der untersten Sytemebene anzugehen und zwar noch bevor das Betriebssystem überhaupt geladen wird. Hier kommt die DSDT ins Spiel die ja bei Notebooks eh in den meisten Fällen umfassend angepasst werden muss (Stichwort Akku Anzeige, DisplayHelligkeitsreglung etc.). Hier jetzt umfassend zu erklären was eine DSDT ist und für was sie im einzelnen gut ist würde den Rahmen sprengen zumal es auch genügend Lesestoff dazu in der FAQ und Themenbereichen im Forum gibt die sich mit dem Thema befassen. Vielleicht nur so viel die Anpassungen in der DSDT sorgen dafür, dass OS-X Informationen zur genutzten Hardware in einem Format erhält die es verstehen kann. Weiterhin bietet diese Methode auch die Möglichkeit ein wenig zu schummeln und zum Beispiel bei der WLAN Karte eine andere DeviceID unterzuschieben. Im gegeben Beispiel fügen wird folgenden Informationen dem WLAN Gerät hinzu:

Code

1. Method (\_DSM, 4, NotSerialized)
2. {
3. If (LEqual (Arg2, Zero)) { Return (Buffer() { 0x03 } ) }
4. Return (Package())
5. {
6. "device-id", Buffer() { 0x30, 0x00, 0x00, 0x00 },
7. "name", "pci168c,30",
8. "AAPL,slot-name", Buffer() { "AirPort" },
9. "device\_type", Buffer() { "AirPort" },
10. "model", Buffer() { "Atheros 9285 802.11 b/g/n Wireless Network Adapter" },
11. "subsystem-id", Buffer() { 0x8F, 0x00, 0x00, 0x00 },
12. "subsystem-vendor-id", Buffer() { 0x6B, 0x10, 0x00, 0x00 },
13. })
14. }

Alles anzeigen

Dem System gegenüber stellt sich jetzt unsere 9285 mit der DeviceID 0030 anstelle von 002b vor und diese DeviceID ist in der Info.plist enthalten. OS-X weiß also auf diese Weise, dass es ein WLAN Gerät von Typ Atheros 9285 mit der VendorID 168c und der DeviceID 0030 im System gibt und bindet die IO80211Family.kext an dieses Gerät. Das wirklich schöne an dieser Lösung ist, dass wir am WLAN Gerät selbst und auch am OS-X selbst nichts verändert haben denn für den Rechner bleibt es bei der Karte mit der DeviceID 2b der Rechner selbst Ignoriert die übergebenen Informationen nämlich vollständig und spricht das Gerät auf Hardware Ebene weiterhin über seine eigentliche Adresse im PCI Adressraum an und auch andere

Betriebssysteme (Windows oder Linux) ignorieren diese Informationen und sprechen das Gerät weiterhin über seine eigentliche Adresse an. Fazit eine nicht ganz triviale dafür aber schön saubere Lösung 😄

Btw. die Credits für den DSDT Patch gehen an RehabMan der ein wirklich gut sortiertes Repository mit allen möglichen Patches speziell für Laptops unterhält.