Project Evil Windows-Treiber in FreeBSD

FreeBSD 5.x / 6.x

Da FreeBSD einer ständigen Weiterentwicklung unterliegt, kann es unter Umständen möglich sein, dass die eine oder andere Komponente in diesem Flyer nicht enthalten ist. In diesem Fall besuchen Sie bitte die Homepage von FreeBSD:

www.freebsd.org

Unter „Latest Releases“ finden Sie einen Hinweis, der Sie zum gewünschten Abschnitt weiterleitet.

Des Weiteren stehen Ihnen die Mitglieder des deutschsprachigen BSD-Forums unter

www.bsdforen.de

gerne mit Rat und Tat zur Verfügung.

Hinweis:

Dieses Dokument ist eine Ergänzung zum Flyer: *WLAN-Netzwerkkarten*.

Project Evil: Windows-Treiber in FreeBSD

Verfasser: David Chisnall, July 2005

Übersetzer: Jürgen Dankoweit, Oktober 2005

Vorstellung „Project Evil“

Eines der Probleme, welche die freie Software-Gemeinde plagen, ist die Verfügbarkeit von Gerätetreibern. Solange ein Betriebssystem keine deutliche Marktdurchdringung hat, macht es keinen ökonomischen Sinn, Gerätetreiber dafür zu entwickeln.

Viele Hersteller stellen noch nicht einmal eine Dokumentation zur Verfügung, die es erlaubt Treiber zu schreiben, weil sie meinen, dass dies eine Aufdeckung geistigen Eigentums bedeuten würde.

Im Fall von WiFi-Karten kann dies zu einem Problem werden. Es ist sehr schwierig zu sagen, welcher Chipsatz zukünftig für eine Karte verwendet wird – manche Hersteller ändern die Hardware ohne die Produktbezeichnung zu modifizieren. So kann die Suche nach einer WiFi-Karte für das bevorzugte Betriebssystem zum Problem werden.

Die FreeBSD-Gemeinde hat daher das „Project Evil“ ins Leben gerufen. Das ist eine teilweise Einbindung des Windows-APIs, welche es erlaubt, Windows-Treiber für Netzwerkkarten zu verwenden.

Wie funktioniert „Project Evil“?

„Project Evil“ stellt Basisfunktionen zur Verfügung, die von Windows Netzwerkkarten-Treibern verwendet werden. Diese Funktionen werden intern in das FreeBSD-Treibermodell umgesetzt. Für den Treiber sieht es so aus, als ob er in einer Windows-Umgebung laufen würde. Aus Sicht des Betriebssystems ist es ein natives FreeBSD-Kernelmodul.

Unter Windows besteht ein WiFi-Treiber aus drei Komponenten. Der Treiber selber, der meist die die Erweiterung „`.sys`“ hat. Es gibt auch eine „`.inf`“-Datei, die Informationen zum Treiber enthält. Schließlich gibt es noch eine Datei mit einer Kopie der Karten-Firmware.

Traditionell wird die Firmware einer Netzwerkkarte in einem ROM gebrannt. Später wurde erkannt, dass eine Möglichkeit einer Aktualisierung geben sein muss, und so wurde die Firmware in einem Flash-ROM gespeichert. In modernen Billigkarten wird kein Flash-ROM mehr verwendet und so wird die Firmware im RAM hinterlegt. Das bedeutet, dass der Treiber diese erst laden muss, bevor die Karte benutzt werden kann.

Um die Sache noch komplizierter zu machen, haben einige Treiber separate Firmware für den Ethernet-Controller und für Funkteil. Firmware-Dateien haben meist die Endung „`.bin`“.

Erstellen der Kernelmodule

Es wird eine Kopie des Windows-Treibers benötigt. Diese befindet sich auf der Treiber-CD des Herstellers oder auf dessen Internet-Seite. Nun alle Dateien mit der Erweiterung „`.sys`“, „`.inf`“ und „`.bin`“ nach
`/sys/modules/if_ndis` kopieren.

Für dieses Tutorial verwendet der Übersetzer eine Wifi-Karte „WIFI1234“, so dass man man die Treibernamen durch die eigenen ersetzen muss. Diese Treiber werden mitgeliefert:

`wifi1234.bin`	Firmware der Netzwerkkarte
`wifi1234funk.bin`	Firmware des Funkteils
`WIFI1234.INF:`	Datei mit Treiberinformation
`wifi1234.sys:`	Der Treiber selber

Die Art und Weise, wie „Project Evil“-Kernelmodule erstellt werden, hat sich von FreeBSD 5.3 zu FreeBSD 5.4 geändert und leider enthält die Dokumentation zu FreeBSD 5.4 noch immer die Beschreibung der Vorgehensweise unter FreeBSD 5.3.

Dieses Tutorial beschreibt aber nur den Weg, der ab FreeBSD 5.4 gültig ist!

Wichtiger Hinweis:

Es ist sinnvoll nach -STABLE zu aktualisieren, da „Project Evil“ ständig weiterentwickelt wird.

Vorgehensweise

Ab FreeBSD 5.4 werden für „Project Evil“ keine Kernel-Quelltexte mehr benötigt. Das „ndis“- und „if_ndis“-Modul sollten bereit installiert sein. Es muss nur noch ein Modul generiert werden, welches den Treiber und die Firmware enthält. Dies wird von dem Assistent *ndisgen* gehandhabt.

```
# ndisgen
```

Dieses Programm fragt nach dem Verzeichnis, in dem Treiber und Firmware abgelegt sind. Bitte beachten, dass zwischen Groß- und Kleinschreibung unterschieden wird und es muss auch der volle Verzeichnisname angegeben werden. Als Ergebnis erhält man ein einzelnes Modul (`.ko`). In diesem Fall ist es `wifi1234_sys.ko` Das Modul ist nur noch nach `/boot/kernel` zu kopieren.

```
# cp wifi1234_sys.ko /boot/kernel/
# kldload ndis
# kldload if_ndis
# kldload wifi1234_sys
```

Auch hier gilt es wieder zu beachten, dass die Reihenfolge der `kldload`-Kommandos sehr wichtig ist! Bei Missachtung droht ein „kernel panic“.

Wie vorherigen Abschnitt beschrieben sollte man auch wieder Eintragungen in `/boot/loader.conf`vornehmen:

```
ndis_load="YES"
if_ndis_load="YES"
wifi1234_sys_load="YES"
```

Nun kann das System neu gestartet werden. Wie bereits beschrieben wird mit `/stand/sysinstall` die Netzwerkkarte konfiguriert.

„Project Evil“ funktioniert sowohl auf Systemen mit 32 Bit oder 64 Bit Prozessoren von AMD oder Intel. Voraussetzung ist, dass die Treiber entweder 32 Bit oder 64 Bit sind!

Zusammengestellt: Juergen.Dankoweit@T-Online.de
Stand der Erhebung: 16. Oktober 2005

Sonstige Informationen

Hinweis:
Ab FreeBSD 6.x ist nur noch der Assistent *ndisgen* zu verwenden!

- Kernel compilieren (i386 !!!)

Natürlich kann man den ndis-Treiber auch in den Kernel ein-compilieren:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC NDISKERNEL
```

Um den FreeBSD-Kernel NDIS-fähig zu machen, müssen zwei Einträge gemacht bzw. auskommentiert werden:

```
options NDISAPI
device ndis
device wlan # natürlich nur, wenn wlan verwendet wird!
```

Speichern der Konfiguration

```
# cd /usr/src
# make buildkernel KERNCONF=NDISKERNEL
# make installkernel KERNCONF=NDISKERNEL
```

NDISKERNEL ist ein Platzhalter für den Namens des eigenen Kernels! Beim Compilieren und Installieren ist unbedingt auf Fehlermeldungen zu achten!

Bitte vergessen Sie vor dem Erstellen eines neuen Kernels nicht, sich ein separates Verzeichnis anzulegen und den alten Kernel und seine Module zu sichern (z.B. /root/kernels/NDISKERNEL).

Durch diese Prozedur spart man sich in der `/boot/loader.conf` die Zeile `if_ndis_load="YES"`

Weiterführende Links

FreeBSD-Homepage: www.freebsd.org

BSD-Forum (deutsch): www.bsdforen.de

Originalartikel (englisch):
www.pingwales.co.uk/tutorials/project-evil.html

Wenn Sie beim Kauf einer WLAN-Karte sicher gehen wollen, dass ein Atheros-Chipsatz integriert ist, dann empfehlen wir einen Blick auf die Atheros-Homepage:
http://customerproducts.atheros.com/customerproducts/ResultsPageBasic.asp.

Hier sind alle Hersteller aufgeführt, die Atheros-Produkte verwenden.

Zusammengestellt: Juergen.Dankoweit@T-Online.de
Stand der Erhebung: 16. Oktober 2005