

OPT ndiswrapper

Christian Kneisel email: christian.kneisel@gmx.de

10.10.2004

1. Dokumentation des optionalen Paketes ndiswrapper	3
1.1 OPT_NDISWRAPPER='yes'	3
1.1.1 Voraussetzung	3
1.1.2 Standardinstallation	3
1.2 NDISWRAPPER_DEBUG='yes'	4
1.3 Erweiterung um neue Chipsaetze	4
1.3.1 Vorbereitungen zum neuen Treiber	5
1.3.1.1 Vorbereitung	5
1.3.1.2 Hardwarekompatibilitaet	6
1.3.1.3 Konfigurationserstellung und Test	7
1.3.1.4 Sichern und erweitern	8
2. Anhang zum optionalen Paket ndiswrapper	8
2.1 FAQ	8
2.2 TODO	9
2.3 Beispiele	9
2.3.1 Shell-Skript /usr/sbin/ndiswrapper	9
2.3.2 Log-Ausgabe waehrend der conf-Erstellung	9
2.3.3 Log-Ausgabe bei Auflistung mit ndiswrapper -l	11
2.3.4 Ausgabe iwconfig	11

Dokumentation des optionalen Paketes ndiswrapper

1.1 OPT_NDISWRAPPER='yes'

Die unterstuetzte WLAN-Chip-Basis fuer fli4l ist schon extrem gross. Leider gibt es aber in der Riege der WLAN-Chiphersteller immer noch Anbieter, die fuer ihre Chipsaetze keine oeffentlich verfuegbaren Linux/UNIX-Treiber bereitstellen (z.B. Intersil Javelin/Xbow, rtl8180L, Broadcom, ...).

Ein Teil dieser Menge an nicht unterstuetzten Chipsaetzen ist durch eigene Linux-Projekte abgedeckt worden, ein Teil hat noch nicht mal dies, sondern nur fuer Microsoft Windows entwickelte Treiber. Diesem Umstand verdankt das Projekt ndiswrapper (<http://ndiswrapper.sourceforge.net/>) seine Existenz.

Es implementiert um die MS-Windows-Treiber einen Wrapper, der zum MS-Windows-Treiber die Schnittstellen der MS-Windows Kernumgebung abbildet (ntoskernel, ndis, hal, etc.) und zum Linux-Kernel diese auf die net, wlan, usb, pci, etc. -Module abbildet.

Dieses Projekt ist mit leichten Aenderungen im Konfigurationsteil bestehend aus dem CVS-Stand ndiswrapper.0.10.rc1 fuer fli4l angepasst.

Die vom ndiswrapper unterstuetzten Chipsaetze sind hier:

<http://ndiswrapper.sourceforge.net/wiki/index.php/List>

einzusehen.

1.1.1 Voraussetzung

Das ndiswrapper-Modul ist gegen die uClibc 0.9.26 gelinkt und somit fuer fli4l 2.1.8 und mit geringen Aenderungen auch fuer fli4l 2.1.7 geeignet.

Das ndiswrapper-Modul kann mit usb-, pci- und pcmcia-Karten umgehen. Die zusaetzlichen OPTs (z.B. OPT-usb) muessen aktiviert sein.

Das Paket enthaelt zum Anfang fuer die Chipsaetze Realtek 8180L und Intersil Javelin die Konfiguration und Treiber. Es ist aber ohne weiteres mit etwas eigener Initiative um weitere unterstuetzte Chipsaetze zu erweitern (z.B. Broadcom etc.). Dazu sind MS-Windows-Treiber noetig, bevorzugt die aktuellsten fuer MS-Windows-XP. Es wird die jeweilige Treiber.inf-Datei und Treiber.sys-Datei benoetigt.

1.1.2 Standardinstallation

Folgende Variablen sind zu setzen:

- config/ndiswrapper.txt

Es gibt in der Datei config/ndiswrapper.txt folgende zu editierende Variablen:

OPT_NDISWRAPPER	aktiviert die Installation des ndiswrapper (yes/no)
-----------------	---

NDISWRAPPER_DEBUG	uebernimmt ein etwas gespraechigeres ndiswrapper-Modul (Achtung, schreibt sehr! viel in das log) (yes/no)
NDISWRAPPER_CARDS_N	Anzahl der ndiswrapper-verwalteten Karten (Numeric)
NDISWRAPPER_NET_%_DRV	Chipsatz der %'ten ndiswrapper-verwalteten Karte. Momentane erlaubte Werte: (rtl8180,prismIII)
NISDRIVER_%_ACTION	Einbau der MS-Windows-XP-Treiber-Dateien im Verzeichnis opt/files/usr/local/wlan/DRIVER/* passend zum Treiber %.

- config/base.txt

In der Datei config/base.txt muss ndiswrapper als letzter NET_DRV_%-Eintrag eingetragen werden (% ist gleich der Variable NET_DRV_N), die dazugehoerige Option bedeutet, dass das ndiswrapper-Modul anstatt der wlan%-Device-Numerierung eth%-Nummern verwendet und somit in den normalen eth-/wlan-Konfigurationsablauf passt. Der Eintrag als letzter NET_DRV_% muss deshalb gelten, damit bei mehreren ndiswrapper-Karten die automatisch fortlaufende eth%-Numerierung dieser Karten bei Initialisierung des Moduls auf die IP_NET_%- und HOST_%-Konfiguration passt. Wird dies nicht beruecksichtigt, schlaegt das Erstellen des fli4l mit einer Fehlermeldung fehl.

```
NET_DRV_%           ='ndiswrapper'
NET_DRV_%_OPTION    ='if_name=eth%d'
```

Mit dem Setzen der Variable OPT_NDISWRAPPER='yes' und der Konfiguration der aufgezaehlten Variablen werden die Modul-, Runlevel- und Konfigurations-Dateien fuer das fli4l-System zusammengestellt und integriert.

1.2 NDISWRAPPER_DEBUG='yes'

Mit dem Debugging wird ein sehr gespraechiges ndiswrapper-Modul installiert. Es empfiehlt sich, diese Ausgaben in eine Datei zu schreiben. Deswegen wird bei Auswahl des Debuggings ueberprueft ob OPT_KLOGD und OPT_SYSLOGD jeweils gesetzt sind und auch in eine Datei (sonst nach '/var/log/ndiswrapper.dbg.log') schreiben.

Das ndiswrapper-Modul mit Debuggingfunktion ist geringfuegig groesser als das normale ndiswrapper-Modul.

1.3 Erweiterung um neue Chipsaetze

Der ndiswrapper ist mit einigen weiteren Chipsaetzen getestet worden und auch im Einsatz.

Fuer das angepasste fli4l-2.1.8-ndiswrapper-Modul gibt es noch keine weiteren Tests mit speziellen Treibern und Karten fuer fli4l ausser den mitgelieferten. Desungeachtet ist der ndiswrapper darauf ausgelegt, leicht weitere mit fli4l-2.1.8 getestete und funktionierende Chipsaetze in dieses Paket mit einzubinden. Folgendes beschreibt das Vorgehen, um selbst erweitern zu koennen.

1.3.1 Vorbereitungen zum neuen Treiber

Das Einbinden eines neuen MS-Windows*-Treibers passend zu einem neuen Chipsatz besteht aus drei Schritten:

1. Ueberpruefung auf Hardwarekompatibilitaet: Der Chip samt Karte muss zur vorhandenen Hardware passen! Manche neueren und schnelleren WLAN-Karten verlangen zwingend eine PCI-Version ≥ 2.2 . Das leistet nicht jedes fli4l-Router-wuerdige Mainboard und jeder PCI-Slot.
2. Erstellen und Einbinden der Konfiguration zu diesem neuen, vom ndiswrapper-Modul zu ladendem Treiber.
3. Sichern und erweitern des ndiswrapper-Opts fuer alle.

Das Paket ndiswrapper unterstuetzt, im folgenden beschrieben, diesen Ausbau vom Build-System aus, um eine platzsparende Installation zu gewaehrleisten. Eine Installation on the fly ist aber auch auf dem Zielsystem moeglich.

1.3.1.1 Vorbereitung

Der jeweils startende fli4l-Router hat mindestens

- NDISWRAPPER_DEBUG='yes' gesetzt und
- die beiden neuen zu der neuen WLAN-Karte mitgelieferten MS-Windows-XP-Treiber-Files mit den Datei-Endungen inf und sys befinden sich im Verzeichnis opt/files/usr/local/wlan/DRIVER. (Die Beschaffung dieser Files aus kompakten MS-Windows-XP-Treibern ist jedem zur Uebung selbst ueberlassen, ein guter Tipp ist <http://ndiswrapper.sourceforge.net/wiki/index.php/List> und das Wiki)
- die Datei 'check/ndiswrapper.exp' ist fuer die Variable NDISWRAPPER_DRIVER um den neuen Chiptyp zu ergaenzen (z.B. 'BMC84309' o.ae, rtl8180 und prismlll gibt es schon)
- die Datei 'check/ndiswrapper.ext' ist zu ergaenzen:
 - duplizieren der Template-Section:


```
#
# template-Section for extending with new driver, please first duplicate this template-Section,
#   remove leading # and fill in the correct Chiptype and MS-Windows-driver-files
#   use correct syntax and avoid misspelling
#
# if ( ndiswrapper_net_%_drv[j] =~ "INSERT_NEW_CHIPTYPE_HERE" && ndiswrapper_%_action[j] =~ "install"
# )
# then
#
#   add_to_opt "files/usr/local/wlan/DRIVER/MS-Windows-driver.inf"
#   add_to_opt "files/usr/local/wlan/DRIVER/MS-Windows-driver.sys"
#   add_to_opt "files/usr/local/wlan/DRIVER/MS-Windows-driver.extra-needed-files"
```

```
# add_to_opt "files/usr/local/wlan/DRIVER/MS-Windows-driver.extra-needed-files"
# add_to_opt "files/usr/local/wlan/DRIVER/MS-Windows-driver.extra-needed-files"
#
# fi
Die template-Section
```

- entfernen der fuehrenden # von 'if' bis 'fi'
- Eintragen des in 'check/ndiswrapper.exp' neu definierten Chiptypes anstelle 'INSERT_NEW_CHIPTYPE_HERE'
- Ersetzen von 'MS-Windows-driver.inf', 'MS-Windows-driver.sys' und 'MS-Windows-driver.extra-needed-files' durch die korrekt geschriebenen und auch schon in dieses Unterverzeichnis von ./opt kopierten MS-Windows-Treiber-Files
- die Datei 'config/ndiswrapper.txt' ist um den neuen Chiptype zur ergaenzen indem die Variable NDISWRAPPER_CARDS_N entsprechend veraendert wird und die Variablen NDISWRAPPER_NET_%_DRV um den neuen Chiptype ergaenzt werden und NDISWRAPPER_%_ACTION unbedingt fuer diesen %-Wert auf 'install' gesetzt wird, damit die Dateien aus dem 'opt/files/usr/local/DRIVER/'-Verzeichnis beruecksichtigt werden.
- die Datei 'opt/ndiswrapper.txt' ist um die evtl schon bekannten Dateien im 'opt/etc/ndiswrapper/{ndiswrapper-generated-file-name}/'-Verzeichnis zu ergaenzen. Als Vorlage kann wieder eine zu kopierenden Template-Zeile benutzt werden.

1.3.1.2 Hardwarekompatibilitaet

Im folgenden werden einige zu testende Schritte aufgezahlt, an denen die Hardwareinkompatibilitaet sich auessert und wie man sie evtl beheben kann. Dabei wird im speziellen auf PCI-Karten eingegangen, USB-Karten sind "da nicht so anfaellig", da durch die USB-Schicht abstrahiert wird insbesondere gilt aber auch hierbei, dass USB-2.0-Devices nicht mit Linux Ver < 2.6 richtig funktionieren muessen.

Start-Probleme:

- Startet der Rechner nicht mit der eingebauten Karte => geht nicht :-(
- Startet der Rechner mit der eingebauten Karte nicht ueber die Hardwareerkennung des Bios hinaus => evtl. falscher PCI-Slot oder Karte im Bios falsch anegtriggert (PCI=edge?) oder Reihenfolgenkonflikt mit anderen PCI-Karten oder IRQ-Konflikt
- Manche WLAN-Karten benoetigen ein einmaliges installierendes Starten unter einer Microsoftinstallation, so obskur es sich auch anhoert.

Fli4I-Probleme:

- fli4I bleibt beim Start beim Einbinden eines anderen vorhergehend funktionierendem Kernelmoduls 'haengen' => Inkompatibilitaet dieses Moduls zur

neuen Hardware oder PCI-Inkompatibilitaet => Modul updaten oder neue Hardware

- fli4l kommt bis zum Konsolen-Prompt. /proc/pci zeigt diese Karte aber nicht im passenden PCI-Zweig auf, ebenso zeigt 'lspci -v' zwar den Hardware-Vendor aber ausserdem auch ein unbestimmtes 'unknown device' zu dieser Karte => PCI-Version zu alt oder './opt/files/usr/local/share/pci.ids' ist nicht mehr aktuell.
- In den Boot-Meldungen stehen IRQ-Konflikte, die auch den IRQ der neuen WLAN-Karte verschieben => moeglichst durch 'PCI-Slot-Karten-Shuffle' beheben.

1.3.1.3 Konfigurationserstellung und Test

Konfigurationserstellung:

Die Konfigurationsdateien und notwendigen Dateien der MS-Windows-XP-Treiberdateien zu jedem neuen Chipsatz werden mit dem shell-script /usr/sbin/ndiswrapper erstellt. Dazu muss das /opt Verzeichnis schreibbar sein.

Wird der fli4l-Router mit der unter 1.3.1.1 Vorbereitung aufgezaehlten Grundeinrichtung gestartet, versucht das detect-script des ndiswrappers schon automatisch den neuen Treiber und die dazugehoerenden conf-Dateien zu installieren. Schlaegt dies fehl, sind die entsprechenden log-Eintraege vorhanden. Die Beschreibung zum ndiswrapper-Shellskript ist unter 2.3.1 Shell-Skript /usr/sbin/ndiswrapper zu finden.

Fuer eine manuelle Konfigurationserstellung wird einfach der Befehl

```
'ndiswrapper -i /usr/local/wlan/DRIVER/{new-driver-inf-file.inf}'
```

am Konsolenprompt ausgefuehrt. Eine typische Ausgabe waehrend der *.conf-Erstellung ist unter '2.3.2 Log-Ausgabe waehrend der conf-Erstellung' zu sehen.

Liefert der Installationsbefehl keine Fehler, so kann das Ergebnis mit dem Befehl

```
'ndiswrapper -l'
```

angezeigt werden. Eine Typische Ausgabe ist:

```
'Installed ndis driver:
  2802w: hardware not present
  rtl8180: hardware present '
```

Das bedeutet, dass die Konfigurationsdateien und Treiber fuer den Chipsatz der Karte 2802w installiert sind, die Hardware aber nicht gefunden wird, im Gegensatz zum Chipsatz der Karte rtl8180, fuer den sogar die Hardware vorhanden ist.

Test:

Der Erfolg der manuellen Konfiguration laesst sich anhand folgender Befehle durch Nachschauen und manuelles laden des ndiswrapper-Modules ueberpruefen zusaetzlich zu den Ausgaben des ndiswrapper-Skriptes:

- 'ls -ltra /etc/ndiswrapper/*' zeigt alle installierten Dateien im Conf-Verzeichnis auf

ist es leer, konnte kein Treiber installiert werden.

- 'lsmod' zeigt alle geladenen Module auf; ist ein ndiswrapper-Modul zu sehen, laesst sich das Modul auf jeden Fall zu einer vorhandenen Konfiguration laden
- 'rmmod ndiswrapper' entlaedt das ndiswrapper-Modul
- 'insmod /lib/modules/2.4.26/kernel/drivers/net/wireless/ndiswrapper.o if_name=eth%d' Versucht das ndiswrapper-Modul zu laden.
- 'iwconfig [eth% [ESSID|NICK|ENC ...|MODE|RATE]]' Zeigt alle wlan-faehige Devices auf. Mit der hoechsten eth%-Id sind im Erfolgsfall die ndiswrapper-devices. Die Parameter in Klammern lassen mit dazugehorenden Werten eine Konfiguration des wlan-faehigen Devices zu.
- 'ifconfig [eth% {IP} up]' Zeigt alle konfigurierten Netzwerk-Devices an, die Parameter in Klammern starten ein angegebenes Netzwerkdevice zur IP

1.3.1.4 Sichern und erweitern

Kann man die Konfiguration erstellen, das ndiswrapper-Modul mit insmod laden, zeigt iwconfig das dazugehoerende Device an, akzeptiert es WLAN-Parameter-Aenderungen und kann sich connecten, so verbleibt fuer eine dauerhafte Installation nur noch das Erstellen und Anpassen des 'detect_500_ndiswrapper'-Skriptes im 'opt/files/usr/local/wlan/'-Verzeichnis und die Uebername der WLAN-Parameter in 'config/wlan.txt' sowie das Sichern und Bereitstellen der angepassten Dateien incl. der *.conf-Dateien (Achtung, diese muessen im Paket mit Nr_Nr_Nr_Nr.conf stehen, sonst kann man das Paket nicht unter MS Windows auspacken!) an den Maintainer dieses OPTs (christian.kneisel@gmx.de) zur Verfuegung fuer alle. Vielen Dank!

2. Anhang zum optionalen Paket ndiswrapper

2.1 FAQ

- Der Rechner bootet nicht mit WLAN-Karte.

Ist es ein alter 486? So. ca. 12-15 Jahre? Dann kann er evtl nicht mit PCI2.1, PCI2.2 umgehen. => Neuerer Rechner oder andere Karte.

- Meine Karte wird momentan nicht in der ausgelieferten Version unterstuetzt, steht aber auf der ndiswrapper-Seite funktionierender Chipsaetze.

Siehe 1.3 Erweiterung um neue Chipsaetze und auch die Datei ./INSTALL im Paket.

- Mein ndiswrapper laedt nicht beim Start. Was soll ich tun?

Als erstes die log-Funktion von fli4l in Datei einschalten (SYSLOGD, KLOGD). Sind die oben angegebenen Dateien vorhanden? Also die Windows-Treiber-Dateien? Die ndiswrapper-Dateien? Passen die windows-Treiber-Dateien? Die Logs untersuchen, warum der ndiswrapper nicht laeuft. Meist passt das Umsystem nicht (fehlende Rechte, fehlende Dateien, etc.) Notfalls auch die Debugversion testen.

- Das Laden des ndiswrapper.o-Moduls liefert einen EXEC/FORMAT-Fehler.

Das loadndisdriver-Programm muss unbedingt in /sbin liegen!

2.2 TODO

Anpassen des NDISWRAPPER and Version ndiswrapper.0.11

2.3 Beispiele

2.3.1 Shell-Skript /usr/sbin/ndiswrapper

Standardausgabe:

```
fli4l 2.1.8 # ndiswrapper
ndiswrapper shell-script, by knc
```

Uebergabe:
action:

Usage: ndiswrapper OPTION

Manage ndis drivers for ndiswrapper.

Parameter:

```
-i infile [-d opt] Install driver described by infile
-e driver [-d opt] Remove driver
-l          List installed drivers
-m          Write configuration for modprobe
```

-d Debug-Option

Dbg-Option:: (all|fct-List)
fct-List:: fct-Name -r

(-r entspricht einem unterdruecktem read pro Debugausgabe!)

moegliche fct-Namen: all main set_Glob_Var isInstalled loadinf getSection InStrings remComment
getKeyVal initStrings isinfuzz addPCIFuzzEntry write_conf open_new_conf sub_str addReg finddir findfile CopyFiles
parseDevice parseVendor parseManu parseID processPCIFuzz remove try_to_install getPresentCards isinstalled list
modconf_err modconf extra trim parseVersion setVersionVal getVersionVal

Der Parameter '-m' ist fuer den Router fli4l ohne Belang, da die modprobe-/ modconf-/ depmod-Utilities nicht vorhanden sind.

Beispiel:

Aufruf fuer Anzeige aller installierter Hardware mit Debugausgabe der Fkt. getPresentCards und der Funktion list mit Unterdrueckung der Bestaetigung pro Debugausgabe fuer getPresentCards:

```
ndiswrapper -l -d getPresentCards -r -d list
```

2.3.2 Log-Ausgabe waehrend der conf-Erstellung

Befehl:

ndiswrapper -i /usr/local/wlan/DRIVER/2802W.inf

Ausgabe:

teste -:i: mit evtl. arg ./tmp/2802W.inf:
ndiswrapper shell-script, by kno

Uebergabe:
action: i

```
try to install with ./tmp/2802W.inf
ab hier wird ./tmp/2802W.inf nach /etc/ndiswrapper installiert!
Pfadname: ./tmp
2802w ist _NICHT_ installiert!
/etc/ndiswrapper existiert!
Installing 2802w
habe /etc/ndiswrapper/2802w erstellt!
Testausgabe: Sections[0]: Header;1-11; 1>&2
Testausgabe: Sections[1]: Version;13-21; 1>&2
Testausgabe: Sections[2]: ControlFlags;23-25; 1>&2
Testausgabe: Sections[3]: Manufacturer;27-28; 1>&2
Testausgabe: Sections[4]: DeviceList.NTx86.5.1;30-31; 1>&2
Testausgabe: Sections[5]: DeviceList;33-35; 1>&2
Testausgabe: Sections[6]: PRISM_DCB1;37-40; 1>&2
Testausgabe: Sections[7]: PRISM_DCB1.NT;42-47; 1>&2
Testausgabe: Sections[8]: PRISM_DCB1.NT.Services;49-50; 1>&2
Testausgabe: Sections[9]: PRISM_DCB1.NT.ColnInstallers;52-54; 1>&2
Testausgabe: Sections[10]: PRISM_DCB1_XP;56-61; 1>&2
Testausgabe: Sections[11]: PRISM_DCB1_XP.Services;63-64; 1>&2
Testausgabe: Sections[12]: PRISM_DCB1_XP.ColnInstallers;66-68; 1>&2
Testausgabe: Sections[13]: PRISM_DCB1.reg;70-71; 1>&2
Testausgabe: Sections[14]: PRISM_DCB1.reg;73-80; 1>&2
Testausgabe: Sections[15]: PRISM_COINSTALL.reg.NT;82-83; 1>&2
Testausgabe: Sections[16]: PRISM_DRIVER_DCB1.Service;85-91; 1>&2
Testausgabe: Sections[17]: PRISM_DRIVER.EventLog;93-94; 1>&2
Testausgabe: Sections[18]: PRISM_DRIVER.EventLog.reg;96-101; 1>&2
Testausgabe: Sections[19]: COMMON_ICB.reg;103-111; 1>&2
Testausgabe: Sections[20]: COMMON_NDIS.reg;113-123; 1>&2
Testausgabe: Sections[21]: COMMON_NDIS.reg.NT;125-127; 1>&2
Testausgabe: Sections[22]: COMMON.reg;129-152; 1>&2
Testausgabe: Sections[23]: COMMON_AMAC_DEL.reg;154-160; 1>&2
Testausgabe: Sections[24]: COMMON_AMAC.reg;162-232; 1>&2
Testausgabe: Sections[25]: COMMON_PSMODE.reg;234-244; 1>&2
Testausgabe: Sections[26]: COMMON_PS_MODE_XP.reg;246-259; 1>&2
Testausgabe: Sections[27]: DestinationDirs;261-270; 1>&2
Testausgabe: Sections[28]: PRISM_CCU.copy;272-273; 1>&2
Testausgabe: Sections[29]: PRISM_DRIVER.copy;275-276; 1>&2
Testausgabe: Sections[30]: PRISM_DRIVER.copy.NT;278-279; 1>&2
Testausgabe: Sections[31]: PRISM_DRIVER.copy.XP;281-282; 1>&2
Testausgabe: Sections[32]: PRISM_FWDCB.copy;284-285; 1>&2
Testausgabe: Sections[33]: PRISM_FWDCB.copy.NT;287-288; 1>&2
Testausgabe: Sections[34]: PRISM_COINSTALL.copy.NT;290-291; 1>&2
Testausgabe: Sections[35]: SourceDisksNames;293-296; 1>&2
Testausgabe: Sections[36]: SourceDisksFiles;298-302; 1>&2
Testausgabe: Sections[37]: SourceDisksFiles.X86;304-309; 1>&2
Testausgabe: Sections[38]: Strings;311-351; 1>&2
der Valstring:echo "11/25/2003, 2.1.13.0"|cut -d "\" -f2
der Valstring:echo "%VER_VENDOR_STR%"|cut -d "\" -f2
[getSection]: Wenn hier, (PRISM_DCB1_XP.NTx86.5.1) nicht gefunden ret=0:-)!
[getSection]: Wenn hier, (PRISM_DCB1_XP.NT) nicht gefunden ret=0:-)!
pdAddreg=PRISM_DCB1.reg, COMMON_ICB.reg, COMMON_NDIS.reg.NT, COMMON.reg, COMMON_PS_MODE_XP.reg,
COMMON_AMAC.reg
pdCopyfiles = PRISM_DRIVER.copy.XP, PRISM_FWDCB.copy.NT, PRISM_CCU.copy
[isinfuzz]: fuzzeintrag(1260:3890) nicht gefunden!
[isinfuzz]: fuzzeintrag(1260:3890) nicht gefunden!
loeschen von /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf ging nicht!
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: NdisVersion|0x50001
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: Environment|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: BusType|5
getVersionVal(DriverVer):11/25/2003,;
getVersionVal(Provider):%VER_VENDOR_STR%;
pdProv:%VER_VENDOR_STR%;pdVer:11/25/2003,;
globTrimLine:%VER_VENDOR_STR%;
pdProvStr:SMC
pdProvStr:SMC:
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: ndis_version|SMC,11/25/2003,
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf:
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf:
```

```

[addReg] III(HKR,Ndi,DeviceID,0,"PCI\VEN_&DEV_&SUBSYS__PRISM_PCI_SUBSYSID_")nix mit NDI\params\*
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: PlatformID|14480
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: VendorDesc|SMC2802W 2.4GHz 54 Mbps
Wireless PCI Adapter
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: 11dMode|0
[ADDREG: ] Erzwingen parameter EnableRadio|0 nach EnableRadio|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: EnableRadio|1
[addReg] III(HKR,NDI ,Service ,0 , "2802W")nix mit NDI\params\*
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: DeviceVxDs|2802W.sys
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: CardBusBridgeLatencyTimer|32
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: CardBusBridgeCacheLineSize|8
[addReg] III(HKR,NDI\Interfaces ,UpperRange ,0 , "ndis5")nix mit NDI\params\*
[addReg] III(HKR,NDI\Interfaces ,LowerRange ,0 , "ethernet")nix mit NDI\params\*
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: PRISMIOCI|0
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: SilentInstall|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: RTSThresh|2347
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: FragThresh|2346
[addReg] III(HKR,defaults,PSMode,0,1)nix mit NDI\params\*
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: PSMode|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: ShortRetryLimit|7
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: LongRetryLimit|4
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: ConfigProfile|256
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: ConfigProfile|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: NitroMode|1
[write_conf] schreibe in /etc/ndiswrapper/2802w/1260:3890:1113:EE03.conf: NitroMode|1
cp /tmp/2802W.sys /etc/ndiswrapper/2802w/2802w.sys
chmod 644 /etc/ndiswrapper/2802w/2802w.sys
cp /tmp/CCU2802.exe /etc/ndiswrapper/2802w/ccu2802.exe
chmod 644 /etc/ndiswrapper/2802w/ccu2802.exe
0

```

2.3.3 Log-Ausgabe bei Auflistung mit ndiswrapper -l

Befehl:

ndiswrapper -l

Ausgabe:

teste -l: mit evtl. arg ::

ndiswrapper shell-script, by kno

Uebergabe:

action: l

list all installed:

getPresentCards card: 10b9:1541

getPresentCards card: 10b9:5243

getPresentCards card: 10b9:5237

getPresentCards card: 10b9:1533

getPresentCards card: 1799:6001

getPresentCards card: 10ec:8029

getPresentCards card: 5333:8a22

getPresentCards card: 10b9:5229

Verkuerzt auf:10b9 1541 ---- 10b9 5243 ---- 10b9 5237 ---- 10b9 1533 ---- 1799 6001 ---- 10ec 8029 ----

5333 8a22 ---- 10b9 5229 ----:

Installed ndis driver:

2802w: hardware not present

rtl8180: hardware present

2.3.4 Ausgabe iwconfig

lo no wireless extensions.

eth0 no wireless extensions.

```

eth1 IEEE 802.11b ESSID:"TESTCK" Nickname:"fli4l"
Mode:Managed Frequency:2.437GHz Access Point: 00:09:5B:B1:59:D2
Bit Rate=11Mb/s Tx-Power:20 dBm Sensitivity=0/3
RTS thr=2432 B Fragment thr=2432 B
Encryption key:AA2F-2A08-0DD7-EC20-5BDC-C2CC-30 Security mode:restricted
Power Management:off
Link Quality:100/100 Signal level:-51 dBm Noise level:-256 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

```

