DHCP-Server Konfigurieren unter CentOS 5.x

Für die Zuweisung der Netzwerkkonfiguration an den Klienten durch unseren Server bedienen wir uns des DHCP[1)], DHCP ist eine *Ergänzung und Erweiterung* von BOOTP[2)]. DHCP wurde im RFC 2131 definiert und bekam von der Internet Assigned Numbers Authority die beiden UDP-Ports 67 und 68 zugewiesen.

Mittels DHCP ist die automatische Einbindung eines neuen Klienten in unser bestehendes Netzwerk ohne große manuelle Konfiguration möglich. Am Klient muss daher nur der automatische Bezug der IP-Adresse eingestellt sein. Beim Start des Klienten am Netz kann dieser die IP-Adresse, die Netzmaske, das Gateway, DNS-Server und weitere Konfigurationsparameter vom DHCP-Server beziehen. Neben diesen klassischen Parametern zählen hierzu auch die Verwendung einer Reihe von weiteren IP-Variablen, wie z.B.: X-Display- , Time-, Swap-, NIS-Server und die Unterstützung von Vendor-Code-Identifiern zum Einsatz im Bereich PXE[3)] finden.

Beim Starten eines Klienten frägt dieser über einen Broadcast im gesamten Netzwerk nach (s)einer IP-Adresse. Als Antwort auf seinen Broadcast erhält er die beiden wichtigsten Parameter:

- IP-Adresse
- Lease-Time

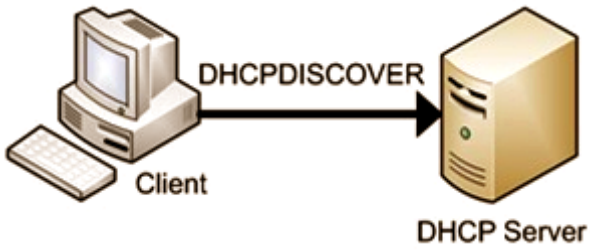
Darüber hinaus können optional noch weitere Parameter mit übergeben werden, wie z.B.:

- Default-Route
- Netzmaske
- DNS-Server-Adressen
- WINS-Server
- Broadcast-Adresse
- IP-Varialen
- sowie noch weitere Parameter

DHCP-Adressvergabe

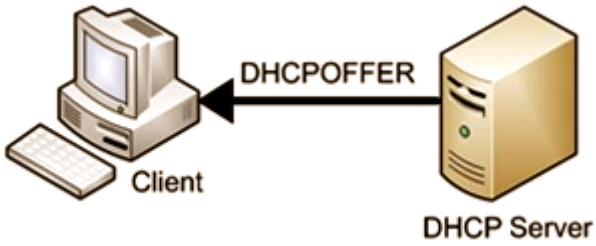
Der grundsätzliche Ablauf bei der Adress-Anfrage folgt dabei folgendem Schema. Die Kommunikation zwischen dem Server (Port 67) und dem Klienten (Port 68) erfolgte mittels UPD[4)].

Beim Booten des Klienten frägt dieser mit einer ***DHCPDISCOVER**-Nachricht* via Broadcast nach seiner Konfiguration. Zu diesem Zeitpunkt besitzt er noch keine eigene IP-Adresse und er kennt auch noch nicht, in welchem Netz er sich befindet. Lediglich seine MAC[5)]-Adresse seines Netzwerkinterfaces ist ihm bekannt. Aus diesem Grund sendet er ein Broadcastpaket mit der Quelladresse 0.0.0.0 und an die Zieladresse 255.255.255.255.



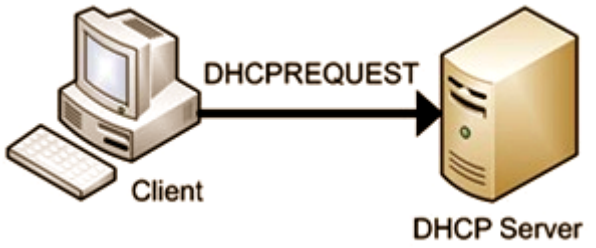

```
Sep 12 21:34:12 nss dhcpd: DHCPDISCOVER from 00:04:13:23:3f:b5 via eth0
```

Dieses Broadcast-Paket beantwortet nun der DHCP-Server mit einer ***DHCPOFFER**-Nachricht*. Das Antwortpaket beinhaltet bereits als Zieladresse die IP, welche der Klient in Zukunft bekommen soll. Da bei der vorherigen Anfrage des Klienten, dieser seine eigene MAC-Adresse mitschickte, kann nun auf diese Weise die DHCPOFFER-Nachricht ihr Ziel finden.



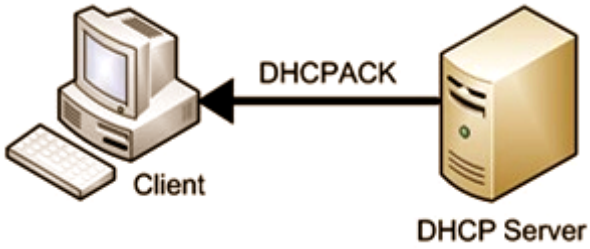

```
Sep 12 21:34:12 nss dhcpd: DHCPOFFER on 192.168.10.61 to 00:04:13:23:3f:b5
via eth0
```

Der Klient hat also vom DHCP-Server ein sogenanntes Angebot (*offer*) bekommen und entscheidet nun, ob es für ihn so in Ordnung ist. Trifft dies zu, sendet er eine ***DHCPREQUEST**-Nachricht*, an den DHCP-Server um diesen mitzuteilen, daß er diese Konfiguration nutzen will.




```
Sep 12 21:34:12 nss dhcpd: DHCPREQUEST for 192.168.10.61 (192.168.10.1)
from 00:04:13:23:3f:b5 via eth0
```

Der DHCP-Server bestätigt dies und sendet eine ***DHCPACK**-Nachricht*, somit besitzt der Klient nun seine eigene IP-Adresse und kennt ggf. noch weitere Parameter für seine weitere Netzwerkkommunkation.




```
 Sep 12 21:34:12 nss dhcpd: DHCPACK on 192.168.10.61 to 00:04:13:23:3f:b5
via eth0
```

Der gesamte erfolgreiche Ablauf aus Sicht des DHCP-Servers entspricht folgendem Diagramm. <uml w=600>

title erfolgreiche Ablauf aus Sicht des DHCP-Servers\n skin BlueModern participant „\n DHCP - SERVER \n“ as links participant „\n Client \n“ as rechts

links ←- rechts : (Port 67) DHCPDISCOVER note right : DHCPDISCOVER mit \nMAC 00:04:13:23:3f:b5 links → rechts : DHCPOFFER (Port 68) note left : DHCPOFFER mit Angabe \nder IP 192.168.10.61 \nan MAC 00:04:13:23:3f:b5 links ←- rechts : (Port 67) DHCPREQUEST note right : DHCPREQUEST mit Angabe \nder IP 192.168.10.61 \nund MAC 00:04:13:23:3f:b5 links → rechts : DHCPACK (Port 68) note left : DHCPACK mit Angabe \nder IP 192.168.10.61 \nund der MAC 00:04:13:23:3f:b5

</uml>

Im **syslog** des DHCP-Servers wird der Ablauf wie folgt festgehalten:

```
Sep 12 21:34:12 nss dhcpd: DHCPDISCOVER from 00:04:13:23:3f:b5 via eth0
Sep 12 21:34:12 nss dhcpd: DHCPOFFER on 192.168.10.61 to 00:04:13:23:3f:b5
via eth0
Sep 12 21:34:12 nss dhcpd: DHCPREQUEST for 192.168.10.61 (192.168.10.1) from
00:04:13:23:3f:b5 via eth0
Sep 12 21:34:12 nss dhcpd: DHCPACK on 192.168.10.61 to 00:04:13:23:3f:b5 via
eth0
```

Sollte die ganze Prozedur Fehl schlagen, z.B. weil der Klient herausgefunden hat, daß die IP-Adresse doppelt vergeben ist, sendet er eine ***DHCPDECLINE**-Nachricht* an der Server. Im Falle einer DHCPDECLINE-Nachricht, sperrt der Server die Adresse für die interne Vergabe und die gesamte Vergabeprozedur beginnt erneut von vorne.

Zusammen mit seiner IP-Adresse erhält der Klient in der DHCPACK-Nachricht auch eine Lease-Time mitgeteilt, welche ihm mitteilt, wie lange die IP-Adresse für ihn reserviert ist. Im RFC Standard wurde definiert, daß der Klient nach der Hälfte der Lease-Time einen erneuten DHCPREQUEST sendet. So teilt er dem Server mit, daß er weiterhin die für ihn reservierte IP-Adresse behalten möchte. Nach Erhalt dieser Nachricht sendet der DHCP-Server eine identische DHCPACK-Nachricht an den Client zurück, in der dann die aktuelle neue Lease-Time mitgeteilt wird. Die IP-Adresse ist somit verlängert und der DCHP-Refresh ist komplett. Sollte der Klient es versäumen eine Verlängerung zu beantragen, muss er die Konfiguration des Netzwerkinterfaces verwerfen und der DHCP-Request startet erneut mit einer DHCPDISCOVER-Nachricht.

Beim Herunterfahren eines Klienten kann dieser dem Server mit einer ***DHCPRELEASE**-Nachricht* den Server informieren, damit dieser die Adresse wieder freigeben kann.

```
 Sep 12 21:58:17 nss dhcpd: DHCPRELEASE of 192.168.10.238 from
00:17:a4:7d:26:1a (hpc6180) via eth0 (found)
```

Der Klient hat aber auch die Möglichkeit, seine zuletzt zugewiesene IP-Adresse über den Reboot hinweg zu „merken“. Dies kann z.B. dann der Fall sein, wenn die Lease-Time, noch nicht abgelaufen ist, oder dem Klienten eine feste IP-Adresse zugeteilt wurde. Dann entfallen die Initialisierungsschritte und der Klient schickt direkt eine ***DHCPREQUEST**-Nachricht* an den DHCP-Server. Dieser bestätigt entweder die Anfrage oder sendet eine ***DHCPNAK**-Nachricht* um dem Klienten mitzuteilen, daß dieser seine gespeicherten Konfigurationen zu löschen, und die Anfrage komplett von vorne zu beginnen hat.

```
 Sep 12 22:01:13 nss dhcpd: DHCPREQUEST for 192.168.10.15 from
00:17:a4:7d:26:1a via eth0
 Sep 12 22:01:13 nss dhcpd: DHCPACK on 192.168.10.15 to 00:17:a4:7d:26:1a
via eth0
```

Installation

Die Installation und Konfiguration des DHCP-Servers gestaltet sich relativ einfach.

Zu erst ist via **yum** der dhcp-Server zu installieren.

```
 yum install dhcp -y
```

Konfiguration

Damit der DHCP-Server richtig funktioniert, muss dessen Konfigurationsdatei noch angepasst werden. Diese hirarchisch aufgebaute config-Datei **dhcpd.conf** liegt unter **/etc**. Zuerst werden allgemein gültige globale Parameter festgelegt, welche für alle Klienten im Netz gelten, z.B. Domainname und Gateway-Adresse.

Als nächstes folgen in dieser Hierarchie die Subnetze. Hier dürfen ebenfalls Parameter für die Klienten festgelegt werden, die sich in diesem Subnetz befidnen. So könnte z.B. jedes Subnetz einen eigenen Default-Router besitzen. Die Parameter innerhalb des Subnetzes überschreiben die global definierten Parameter! Bei solchen Subnetzen ist es wichtig, daß der DHCP-Server nur auf Anfragen aus dem Subnetz antwortet, welche in der dhcpd.conf definiert wurden. Wurde ein Subnetz nicht beschrieben, so werden Anfrage einfach ignoriert.

Die nächste Hierarchiestufe ist der Pool, dieser wird innerhalb eines Subnetzes angelegt. In so einem Pool können auch Bereiche angelegt werden, so können z.B. auch mehrere Pools in einem Subnetz existieren.

Als letzte Stufe in der Hierarchie gibt es die Host-Stufe. In dieser Stufe können einzelne Rechner konfiguriert werden, wenn z.B. diese immer die gleiche IP bekommen sollen. Diese *festen IP-Adressen* widersprichen zwar der Grundüberlegung von DHCP. Aber manchmal kann dies wünschenswert sein, wenn z.B. Zugriffsregeln auf andere Hosts oder Server (Paketfilter oder TCP-Wrapper Regeln) für eine bestimmte IP festgelegt wurden.

Das „group“ Statement hilft dabei, neben der Unterteilung nach Subnetzen, Konfigurationsblöcke mit gleichen Parametern zusammenzufassen. So muss nicht für jeden Host die gesamte Palette wiederholt werden.

Vergebene Adressen werden in der Datei **/var/lib/dhcpd/dhcpd.leases** gespeichert.

Optimierung bei mehreren Netzwerkkarten

Sind im Server mehrere Netzwerkkarten vorhanden, binden wir den dhcp-Server an ein Interface. Dazu tragen wir in der **/etc/sysconfig/dhcpd** folgende Option ein:

```
# Command line options here
DHCPDARGS=eth0
```

Konfigurationsdatei bearbeiten

Anschließend wird die Konfigurationsdatei unter **/etc/dhcpd.conf** entsprechend den eigenen Anforderungen angelegt.

```
ddns-update-style interim;
ignore client-updates;

subnet 192.168.100.0 netmask 255.255.255.0 {

# --- default gateway
        option routers                  192.168.100.100;
        option subnet-mask              255.255.255.0;

        option nis-domain               "nausch.org";
        option domain-name              "nausch.org";
        option domain-name-servers      192.168.100.100;

        option time-offset              -18000; # Eastern Standard Time
#       option ntp-servers              192.168.100.1;
#       option netbios-name-servers     192.168.100.1;
# --- Selects point-to-point node (default is hybrid). Don't change this
unless
# -- you understand Netbios very well
#       option netbios-node-type 2;

        range dynamic-bootp 192.168.100.200 192.168.100.254;
        default-lease-time 21600;
        max-lease-time 43200;

        # we want the nameserver to appear at a fixed address
        host ns {
                next-server marvin.redhat.com;
                hardware ethernet 12:34:56:78:AB:CD;
```

```
                fixed-address 207.175.42.254;
        }
  host laptop
  { hardware ethernet 00:90:46:1b:e4:74;
    fixed-address 192.168.100.51;
  }
  host GXP-2000
  { hardware ethernet 00:0b:62:68:60:bd;
    fixed-address 192.168.100.52;
  }

  host compaq-evo
  { hardware ethernet 00:01:05:55:f2:4b;
    fixed-address 192.168.100.53;
  }

  host compaq-deskpro
  { hardware ethernet 00:15:33:55:FC:A9;
    fixed-address 192.168.100.54;
  }

  host gericom
  { hardware ethernet 00:0b:1a:3c:55:28;
    fixed-address 192.168.100.55;
  }

  host snom360
  { hardware ethernet 00:04:31:32:3f:b5;
    fixed-address 192.168.100.56;
  }

  host snom320
  { hardware ethernet 00:04:31:42:42:2e;
    fixed-address 192.168.100.57;
  }

  host spa2100
  { hardware ethernet 00:01:3e:6a:78:0f;
    fixed-address 192.168.10.58;
  }

}
```

DHCP-Konfiguration überprüfen

Bevor wir nun unseren DHCP-Server das erste mal starten, überprüfen wir unsere Konfiguration mit:

```
# service dhcpd configtest
Syntax: OK
```

DHCP-Server starten

Den ersten Start unseres DHCP-Server nehmen wir wie folgt vor.

```
service dhcpd start
dhcpd starten:                                             [  OK  ]
```

Im syslog wird der erfolgreiche Start entsprechend quittiert:

```
Nov  7 19:06:03 mnss dhcpd: Internet Systems Consortium DHCP Server V3.0.5-
RedHat
Nov  7 19:06:03 nss dhcpd: Copyright 2004-2006 Internet Systems Consortium.
Nov  7 19:06:03 nss dhcpd: All rights reserved.
Nov  7 19:06:03 nss dhcpd: For info, please visit
http://www.isc.org/sw/dhcp/
Nov  7 19:06:03 nss dhcpd: WARNING: Host declarations are global.  They are
not limited to the scope you declared them in.
Nov  7 19:06:03 nss dhcpd: Wrote 0 deleted host decls to leases file.
Nov  7 19:06:03 nss dhcpd: Wrote 0 new dynamic host decls to leases file.
Nov  7 19:06:03 nss dhcpd: Wrote 0 leases to leases file.
Nov  7 19:06:03 nss dhcpd: Listening on
LPF/eth0/00:1f:d0:8c:72:77/192.168.100/24
Nov  7 19:06:03 nss dhcpd: Sending on
LPF/eth0/00:1f:d0:8c:72:77/192.168.100/24
Nov  7 19:06:03 nss dhcpd: Sending on   Socket/fallback/fallback-net
```

Beim Starten eines Klienten, in unserem Faller der Host **laptop** mit der MAC-Adresse **00:90:46:1b:e4:74**, wird das Aushandeln der IP-Adresse vermerkt:

```
Nov  7 19:07:50 mnss dhcpd: DHCPDISCOVER from 00:90:46:1b:e4:74 via eth0
Nov  7 19:07:50 mnss dhcpd: DHCPOFFER on 192.168.100.51 to 00:90:46:1b:e4:74
via eth0
Nov  7 19:07:50 mnss dhcpd: DHCPREQUEST for 192.168.100.51 (192.168.10.2)
from 00:90:46:1b:e4:74 via eth0
Nov  7 19:07:50 mnss dhcpd: DHCPACK on 192.168.100.51 to 00:90:46:1b:e4:74
via eth0
```

automatisches Starten des Dienste beim Systemstart

Damit nun unser DHCP-server beim Booten automatisch gestartet wird, nehmen wir noch folgende Konfigurationsschritte vor.

```
chkconfig dhcpd on
```

Anschließend überprüfen wir noch unsere Änderung:

```
chkconfig --list | grep dhcpd
```

```
dhcpd           0:Aus   1:Aus   2:Ein   3:Ein   4:Ein   5:Ein   6:Aus
```

Überwachung der ARP Tabelle mit arpwatch

Mit dem Netzwerktool ***arpwatch*** können wir die MAC-Adressen mit den zugehörigen IP-Adressen im lokalen Netzwerk überwachen. Es setzt die Netzwerkkarte in den Promiscuous Mode und überwacht die ARP-Pakete, welche das lokale Netz passieren. ***arpwatch*** ist jedoch kein IDS[6)]-Programm, sondern ein Hilfsprogramm, das sich auf eine spezielle Art von Angriffen spezialisiert hat. So ist ***arpwatch*** nur in der Lage, Angriffe auf das ARP-Protokoll zu erkennen (z. B. ARP-Spoofing). Ausserdem erkennt es neu im Netz auftauchende Rechner.

arpwatch wird bei Bedarf einben Alarm auslösen, wenn einer der folgenden Zustände bei der Überwachung der ARP-IP Tabelle auftritt.

Meldung	**Beschreibung**
New Activity	Das Adress Paar wird sechs Monate oder mehr wieder benutzt
New Station	Diese MAC Adresse wurde zum ersten Mal beobachtet
Changed MAC Address	Die MAC Adresse hat sich geändert
Flip Flop	Die MAC Adresse hat sich zu einer schon mal verwendeten MAC Adresse geändert

Tritt mindestens eines dieser Ereignisse auf, kann der **BOfH**[7)] unmittelbar via eMail gewarnt und informiert werden. Ferner werden im Syslog folgende Ereignisse protokolliert:

Meldung	**Beschreibung**
MAC broadcast	Die MAC Adresse des Rechners ist die Broadcast Adresse oder besteht nur aus Nullen
ip broadcast	Die IP-Adresse des Hosts ist eine Broadcast Adresse
Bogon	Die IP-Adresse des Hosts gehört nicht in dieses Subnetz
MAC mismatch	Die Quell MAC Adresse entspricht nicht der MAC Adresse innerhalb des ARP Pakets
reused old MAC address	Die MAC Adresse hat sich auf eine MAC Adresse geändert, die entweder der drittletzten oder noch älteren MAC Adresse entspricht

Mit diesem Überwachungs-Programm können wir nun sehr schnell doppelte MAC- und IP-Adressen, oder weitere Netzwerkfehlkonfigurationen wie auch Angriffe von aktiven Sniffern erkennen.

In der Datei ***/var/arpwatch/arp.dat*** legt **arpwatch** die Adresspaare ab, die es bisher im Netzwerk beobachtet hat.

```
0:4:13:2a:b:6b  192.168.100.174   1254422914      snom300-3
0:4:13:40:3:35  192.168.100.170   1254422890      snom820
0:4:13:23:3f:b5 192.168.100.171   1254422865      snom360
0:4:13:25:c:90  192.168.100.173   1254422441      snom300-2
0:9:45:40:f2:b3 192.168.100.176   1254422688      ST-100
```

Installation

Die Installation unter CentOS geht am einfachsten via **yum**.

```
# yum install arpwatch
```

Was uns das Programmpaket mitbringt, erkunden wir mit der Option **iql** beim Programm **rpm**.

```
# rpm -iql arpwatch
```

```
Name        : arpwatch                     Relocations: /usr
Version     : 2.1a13                            Vendor: CentOS
Release     : 21.el5                     Build Date: Mi 21 Jan 2009
05:13:31 CET
Install Date: Do 01 Okt 2009 20:25:38 CEST      Build Host:
builder16.centos.org
Group       : Applications/System           Source RPM:
tcpdump-3.9.4-14.el5.src.rpm
Size        : 472793                           License: BSD
Signature   : DSA/SHA1, Mo 09 Mär 2009 02:45:17 CET, Key ID a8a447dce8562897
URL         : http://www.tcpdump.org
Summary     : Netzwerküberwachungs-Werkzeuge zum Verfolgen von IP-Adressen
in Netzwerken.
Description :
Das arpwatch Paket enthält arpwatch und arpsnmp.  Arpwatch und
arpsnmp sind beide Netzwerküberwachungswerkzeuge.  Beide Dienstprogramme
überwachen Ethernet oder FDDI Netzwerkverkehr und bauen Datenbanken von
Ethernet/IP
Adressenpaaren, und können von Änderungen via E-mail berichten.

Installieren Sie das arpwatch Paket, wenn Sie Netzwerküberwachungsgeräte,
welche automatisch IP Adressen auf Ihrem Netzwerk im Auge
behalten brauchen.
/etc/rc.d/init.d/arpwatch
/etc/sysconfig/arpwatch
/usr/sbin/arpsnmp
/usr/sbin/arpwatch
/usr/share/doc/arpwatch-2.1a13
/usr/share/doc/arpwatch-2.1a13/CHANGES
/usr/share/doc/arpwatch-2.1a13/README
/usr/share/man/man8/arpsnmp.8.gz
/usr/share/man/man8/arpwatch.8.gz
/var/arpwatch
/var/arpwatch/arp.dat
/var/arpwatch/arp2ethers
/var/arpwatch/arpfetch
/var/arpwatch/d.awk
/var/arpwatch/duplicates.awk
/var/arpwatch/e.awk
```

```
/var/arpwatch/ethercodes.dat
/var/arpwatch/euppertolower.awk
/var/arpwatch/massagevendor
/var/arpwatch/massagevendor-old
/var/arpwatch/missingcodes.txt
/var/arpwatch/p.awk
```

Konfiguration

Viel gibt es nicht zu Konfigurieren, lediglich in der Datrei **/etc/sysconfig/arpwatch**, können wir angeben unter welchen User **arpwatch** laufen, wer die eMail bekommen und wer als Absender benutzt werden soll.

```
 # vim /etc/sysconfig/arpwatch
```

```
# -u <username> : defines with what user id arpwatch should run
# -e <email>    : the <email> where to send the reports
# -s <from>     : the <from>-address
OPTIONS="-u pcap -e root -s 'root (Arpwatch)'"
```

Programmstart

erster Programmstart

Den **arpwatch** daemon starten wir einfach mit dem Aufruf:

```
 # service arpwatch start
 arpwatch starten:                                          [  OK  ]
```

Der erfolgreiche Programmstart wird uns im syslog dokumentiert:

```
Oct  1 20:33:37 nss kernel: device eth0 entered promiscuous mode
Oct  1 20:33:37 nss arpwatch: listening on eth0
Oct  1 20:33:39 nss arpwatch: new station 192.168.100.164 0:4:13:2a:4b:6b
Oct  1 20:33:41 nss arpwatch: new station 192.168.100.245 0:e:8:ec:4a:f6
Oct  1 20:34:23 nss dhcpd: DHCPDISCOVER from 00:04:13:23:3f:b5 via eth0
Oct  1 20:34:23 nss dhcpd: DHCPOFFER on 192.168.100.161 to 00:04:13:23:3f:b5
via eth0
Oct  1 20:34:23 nss dhcpd: DHCPREQUEST for 192.168.100.161 (192.168.100.1)
from 00:04:13:23:3f:b5 via eth0
Oct  1 20:34:23 nss dhcpd: DHCPACK on 192.168.100.161 to 00:04:13:23:3f:b5
via eth0
Oct  1 20:34:25 nss arpwatch: new station 192.168.100.161 0:4:13:23:3f:b5
Oct  1 20:34:27 nss arpwatch: new station 192.168.10.244 0:e:8:eb:5e:2a
```

automatischer Programmstart

Damit nun unser **arpwatch**-Daemon beim Booten automatisch gestartet wird, nehmen wir noch folgende Konfigurationsschritte vor.

```
# chkconfig arpwatch on
```

Anschließend überprüfen wir noch unsere Änderung:

```
# chkconfig --list | grep arpwatch
arpwatch        0:Aus   1:Aus   2:Ein   3:Ein   4:Ein   5:Ein   6:Aus
```

Status-eMail

Beschafft sich nun ein Klient von unserem DHCP-Server eine Adresse, so wird nunmehr diese Aktion an den Sys-Admin per eMail gemeldet:

```
   From: root@nausch.org (Arpwatch)
     To: root@nausch.org
Subject: new station (ST-100.nausch.org)
   Date: Thu,  1 Oct 2009 20:43:05 +0200 (CEST)

           hostname: ST-100.nausch.org
         ip address: 192.168.10.66
    ethernet address: 0:9:45:40:f2:b3
     ethernet vendor: Palmmicro Communications Inc
           timestamp: Thursday, October 1, 2009 20:43:05 +0200
```

Links

- **Zurück zu Projekte und Themenkapitel**
- **Zurück zur Startseite**

[1)]
Dynamic **H**ost **C**onfiguration **P**rotocol
[2)]
Bootstrap Protocol
[3)]
Preboot e**X**ecution **E**nvironment
[4)]
Unreliable **D**atagram **P**rotocol
[5)]
Media **A**ccess **C**ontrol
[6)]
Intrusion **D**etection **S**ystem, ein Computer-Programm zur Erkennung von Angriffen auf Computernetzwerke
[7)]

Bastard **O**perator **F**rom **H**ell