

# WLAN-Access-Point mit hostapd und USB-Stick

Aus DebianforumWiki

Wiki < Netzwerk < WLAN-Access-Point mit hostapd und USB-Stick

Auf der Basis von Wheezy wird hier das Einrichten eines WLAN-Accesspoints vorgestellt, wobei auf eine Installation **ohne Desktop** eingegangen wird.



**Getestet:** Dieser Hinweis soll dir dabei helfen, zu entscheiden, ob dieser Artikel auf deinem System funktionieren wird oder nicht. Solltest du feststellen, dass dieser Artikel bei einer ungetestet Version funktioniert, kannst du das gerne hier korrigieren oder im Forum (<http://debianforum.de/forum/viewforum.php?f=36>) anmerken.

- Dieser Artikel wurde für Debian Wheezy 7.0 getestet.

## Inhaltsverzeichnis

- 1 Einführung
  - 1.1 Benötigte Software
- 2 Installation
  - 2.1 DHCP-Server
    - 2.1.1 /etc/dnsmasq.conf
  - 2.2 hostapd
    - 2.2.1 /etc/hostapd.conf
  - 2.3 Netzwerk-Konfiguration
    - 2.3.1 /etc/network/interfaces
- 3 Inbetriebnahme
- 4 Weblinks
- 5 Quellenangaben

## Einführung

Der Accesspoint unterstützt WPA/WPA2 und wird über eine einfache Firewall (🌀 iptables (<https://packages.debian.org/iptables>) ) abgesichert, die auch das Port-Forwarding und NAT implementiert. In diesem Beispiel wird für die Konfiguration von iptables vorausgesetzt, dass das Kabelnetzwerk eine IP aus dem Netz **192.168.0.0/24** und eine Route ins Internet hat.

Vorhanden sein muss eine Kabelnetzwerkkarte (eth0), sowie ein WLAN-Adapter (wlan0) der den AP-Modus unterstützt. Ob der AP-Modus unterstützt wird, kann mit folgendem Befehl abgefragt werden:

```
root@debian:~$ iw list | grep AP$
* AP
```

Die Ausgabe oben besagt, der AP-Modus wird unterstützt. Alles andere deutet daraufhin, dass der AP-Modus nicht beherrscht wird.

Der WLAN-Adapter in dieser Anleitung ist ein *Ralink Technology, Corp. RT5370 Wireless Adapter* USB-Stick, der vom Modul **rt2800usb** angesteuert wird und die Firmware **rt2870.bin** aus dem Paket  **firmware-ralink** (<https://packages.debian.org/firmware-ralink>) benötigt. Ist die Firmware installiert, funktioniert der USB-Stick out-of-the-box und beherrscht auch den AP-Modus. Hier die genauen Informationen vom USB-System über den Stick:

```
user@debian:~$ lsusb
Bus 001 Device 002: ID 148f:5370 Ralink Technology, Corp. RT5370 Wireless Adapter
```

## Benötigte Software

-  **hostapd** (<https://packages.debian.org/hostapd>)
-  **dnsmasq** (<https://packages.debian.org/dnsmasq>)
-  **wireless-tools** (<https://packages.debian.org/wireless-tools>)
-  **firmware-ralink** (<https://packages.debian.org/firmware-ralink>)
-  **iw** (<https://packages.debian.org/iw>)

Der Accesspoint wird mit der Software  **hostapd** (<https://packages.debian.org/hostapd>) betrieben und für die Verteilung der IP-Adressen an die WLAN-Clients ist  **dnsmasq** (<https://packages.debian.org/dnsmasq>) zuständig. Die Konfiguration erfolgt hauptsächlich in der Datei **/etc/network/interfaces**, sowie in den Konfigurationsdateien von  **dnsmasq** (<https://packages.debian.org/dnsmasq>) und  **hostapd** (<https://packages.debian.org/hostapd>) .

Zusätzlich zu den altbekannten  **wireless-tools** (<https://packages.debian.org/wireless-tools>) , kann das moderne  **iw** (<https://packages.debian.org/iw>) zum Konfigurieren des WLAN-Gerätes verwendet werden.

## Installation

Zunächst wird die benötigte Software installiert, wobei für die Firmware der Eintrag "non-free" in der "sources.list" notwendig ist:

```
root@debian:~$ apt-get install firmware-ralink hostapd wireless-tools dnsmasq iw
```

## DHCP-Server

Die Konfigurationsdatei von  **dnsmasq** (<https://packages.debian.org/dnsmasq>) wird beiseite gelegt und eine angepasste Version erstellt:

```
root@debian:~$ mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
```

### /etc/dnsmasq.conf

```
interface=wlan0
no-dhcp-interface=eth0
dhcp-range=interface:wlan0,10.0.0.20,10.0.0.30,infinite
```

Der erste Eintrag legt wlan0 als DHCP-Interface fest, der zweite schliesst das Kabelnetzwerk eth0 vom DHCP aus, wir wollen nur auf wlan0 IP-Adressen verteilen. Der dritte Eintrag legt die für DHCP verfügbaren IP-Adressen (von/bis) fest.

## hostapd

Der  hostapd (<https://packages.debian.org/hostapd>) erfordert erheblich mehr Konfigurationsaufwand. Die Konfigurationsdatei ist "/etc/hostapd.conf". Falls die Konfiguration nicht funktioniert, sollte man die Startdatei "/etc/init.d/hostapd" überprüfen und gegebenenfalls den Wert DAEMON\_CONF anpassen.

```
DAEMON_CONF=/etc/hostapd.conf
```

Für die Eiligen unter den Lesern hier zunächst eine Konfigurationsdatei für einen Accesspoint mit WPA2 und Pre-Shared-Key. Viele Optionen sind hier in den Standardwerten aktiv. Die Einträge sind entsprechend kommentiert.

### /etc/hostapd.conf

```
# USB-Stick oder WLAN-Gerät
interface=wlan0
# Netlink-Treiber laden
driver=nl80211
# AP-Name
ssid=hostapd1
# Kanal
channel=1
# bis 54Mbit, unterstützt auch 802.11 und 802.11b
hw_mode=g
# Authentifizierungsmodus "Open System Authentication"
auth_algs=1
# akzeptiert nur WPA2-Verbindungen
wpa=2
# Pre-Shared-Key (PSK) im Klartext
wpa_passphrase=secret_word
```

Eine detailliertere Konfigurationsdatei mit Kommentaren hilft bei der Anpassung an spezielle Bedürfnisse. Hier bietet der Accesspoint WPA und WPA2 an, es wird ebenfalls ein Pre-Shared-Key verwendet, die anderen Authentifizierungsverfahren sind ein eigenes Thema.

```
## Log-Einstellungen
# "-1" = alle Module loggen, "3" z.B. loggt nur WPA
logger_syslog=-1
# "2" = informative Meldungen, "1" = Debuggen, "0" = echt alles loggen
logger_syslog_level=2
# wie oben, nur Ausgabe auf Konsole, wenn Programm nicht im Hintergrund läuft
logger_stdout=-1
# wie oben
logger_stdout_level=2

## Statusinformationen
# temporäre Datei für Statusinformationen
dump_file=/tmp/hostapd
# Verzeichnis für Socket für hostapd_cli und andere externe Kontrollprogramme
ctrl_interface=/var/run/hostapd

## SSID und Kanal
# Name des Accesspoints
ssid=hostapd1
# Kanal
channel=1

## Wireless Modus
# g = IEEE 802.11g, Default: IEEE 802.11b
hw_mode=g
```

```
## Intervall für Rundspruch
# Rundspruch (Beacon) "Ich bin ein AP!" in Millisekunden (1 - 65535)
beacon_int=100

## DTIM-Ankündigung
# Ein DTIM ist eine Art Weckruf an die Stationen: "Gleich wird es interessant!".
# Es reicht jedes zweite Beacon mit dem DTIM zu bestücken.
dtim_period=2

## Maximale Anzahl an Clients
# Limit: 2007, man sollte aber realistisch bleiben...
max_num_sta=20

## RTS/FRAGM abschalten
# Das übernimmt das WLAN-Modul, ja, 2347=off
rts_threshold=2347
# und 2346=off
fragm_threshold=2346

## MAC-basierte Authentifizierung
# 0 = akzeptieren, wenn nicht explicit abgelehnt (deny_mac_file)
# 1 = ablehnen, wenn nicht explicit erlaubt (accept_mac_file)
# 2 = RADIUS-Server verwenden, nach Durchsicht der obigen Listen
# Zum Sperren/Erlauben diese Dateien anlegen und bestücken:
# accept_mac_file=/etc/hostapd.accept
# deny_mac_file=/etc/hostapd.deny
macaddr_acl=0

## Authentifizierung
# 1 = Open System, 2= Shared Key --> WEP
# 3 = beides
auth_algs=1

## WPA/WPA2
# 1 = WPA, 2 = WPA2, 3 = beides
wpa=3
# WLAN-Key (Pre-shared Key)
wpa_passphrase=secret_word
# Pre-shared-Key Management
wpa_key_mgmt=WPA-PSK
# WPA2 Verschlüsselung anbieten
rsn_pairwise=CCMP
# WPA Verschlüsselung anbieten
wpa_pairwise=TKIP CCMP

## Schlüsselerneuerung
# Schlüsselteile nach 10 Min. erneuern
wpa_group_rekey=600
# s.o.
wpa_ptk_rekey=600
# erneuern nach 24 Std.
wpa_gmk_rekey=86400

## Gerät und Treiber
# USB-Stick einbinden
interface=wlan0
# An Netlink-Interface übergeben
driver=nl80211
# SSID nicht "verheimlichen"
# 1 = Sende leere SSID und ignoriere Anfragen die die SSID nicht enthalten
# 2 = Sende leere SSID, aber setze die Länge nicht auf 0, sondern genau auf die Länge der SSID,
# einige Clients brauchen das.
ignore_broadcast_ssid=0

## Ländercode
# Ländercode setzen
country_code=DE
# Ländercode aktivieren, "0" = aus (default)
ieee80211d=1
```

Der Ländercode sollte übrigens auch in der Datei `"/etc/default/crda"` gesetzt werden (`REGDOMAIN=DE`).

### Welche Hardware benötigt welche Software-Schnittstelle

- **driver=hostap** für Geräte mit Prism2/2.5/3 Chipsatz.
- **driver=madwifi** für Geräte mit Atheros (ar521x) Chipsatz.

- **driver=nl80211** für Geräte die mit einem mac80211-basierten Treiber funktionieren und den AP-Modus unterstützen.

## Netzwerk-Konfiguration

### /etc/network/interfaces

```
# The loopback network interface
auto lo
    iface lo inet loopback

# The primary network interface
auto eth0
    iface eth0 inet dhcp

# WLAN mit fester Adresse für hostapd
auto wlan0
    iface wlan0 inet static
    address 10.0.0.1
    netmask 255.255.255.0
    broadcast 10.0.0.255

# Firewall zurücksetzen, Tabellen leeren
up /sbin/iptables -F
up /sbin/iptables -X
up /sbin/iptables -t nat -F

# Kabelnetzwerk maskieren, Port-Forwarding sowie Nat aktivieren
up iptables -A FORWARD -o eth0 -i wlan0 -s 192.168.0.0/24 -m conntrack --ctstate NEW -j ACCEPT
up iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
up iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
up sysctl -w net.ipv4.ip_forward=1

# hostapd und dnsmasq (neu) starten
up service hostapd restart
up service dnsmasq restart
```

## Inbetriebnahme

Jetzt wird mit diesem Befehl das ganze Konstrukt gestartet:

```
root@debian:~$ service networking restart
```

Funktioniert es nicht, zunächst den hostapd debuggen:

```
root@debian:~$ hostapd -dd /etc/hostapd.conf
```

Wenn alles ohne Fehlermeldung startet, kann sich mit einem Client verbunden werden. Hierzu **auf einem WLAN-Client** die Datei **wpa\_supplicant.tmp** erstellen.

```
root@debian:~$ wpa_passphrase hostapd1 "secret_word" >> wpa_supplicant.tmp
```

Jetzt den wpa\_supplicant im Debug-Modus starten:

```
root@debian:~$ wpa_supplicant -Dwext -cwpa_supplicant.tmp -iwlan0 -dd
```

Bleibt die Ausgabe irgendwann stehen und sieht ähnlich aus wie das folgende Beispiel, sollte alles funktionieren. Scrollt die Ausgabe immer weiter, stimmt noch irgendetwas nicht.

```
wlan0: WPA: Installing GTK to the driver (keyidx=2 tx=0 len=32)
WPA: RSC - hexdump(len=6): 00 00 00 00 00 00
wpa_driver_wext_set_key: alg=2 key_idx=2 set_tx=0 seq_len=6 key_len=32
wlan0: WPA: Key negotiation completed with 00:1b:11:c8:bd:2b [PTK=CCMP GTK=TKIP]
wlan0: Cancelling authentication timeout
wlan0: State: GROUP_HANDSHAKE -> COMPLETED
wlan0: CTRL-EVENT-CONNECTED - Connection to 00:1b:11:c8:bd:2b completed [id=0 id_str=]
wpa_driver_wext_set_operstate: operstate 0->1 (UP)
netlink: Operstate: linkmode=-1, operstate=6
EAPOL: External notification - portValid=1
EAPOL: External notification - EAP success=1
EAPOL: SUPP_PAE entering state AUTHENTICATING
EAPOL: SUPP_BE entering state SUCCESS
EAP: EAP entering state DISABLED
EAPOL: SUPP_PAE entering state AUTHENTICATED
EAPOL: Supplicant port status: Authorized
EAPOL: SUPP_BE entering state IDLE
EAPOL authentication completed successfully
RTM_NEWLINK: operstate=1 ifi_flags=0x11043 ([UP][RUNNING][LOWER_UP])
RTM_NEWLINK, IFLA_IFNAME: Interface 'wlan0' added
WEXT: if_removed already cleared - ignore event
EAPOL: startWhen --> 0
EAPOL: disable timer tick
```

Ist alles gut holt man sich (auf einer anderen Konsole) eine IP:

```
root@debian:~$ dhclient wlan0
```

Natürlich ist auch über den grafischen Netzwerk-Manager der Desktop-Umgebung und die diversen anderen Tools das Verbinden möglich. Für die Fehlersuche ist die genannte Verbindungsmethode über `wpa_supplicant` aber die erste Wahl. Auf der anderen Seite im Accesspoint sieht das Loggen einer erfolgreichen Verbindung so aus:

```
root@debian:~$ tail -f /var/log/syslog
Sep  8 12:30:26 APTtest hostapd: wlan0: STA 00:14:d1:54:33:42 IEEE 802.11: authenticated
Sep  8 12:30:26 APTtest hostapd: wlan0: STA 00:14:d1:54:33:42 IEEE 802.11: associated
(aid 1)
Sep  8 12:30:26 APTtest hostapd: wlan0: STA 00:14:d1:54:33:42 RADIUS: starting
accounting session 522CA4FD-00000001
Sep  8 12:30:26 APTtest hostapd: wlan0: STA 00:14:d1:54:33:42 WPA: pairwise key
handshake completed (RSN)
Sep  8 12:30:28 APTtest dnsmasq-dhcp[1729]: DHCPREQUEST(wlan0) 10.0.0.20
00:15:d2:55:33:44
Sep  8 12:30:28 APTtest dnsmasq-dhcp[1729]: DHCPACK(wlan0) 10.0.0.20 00:15:d2:55:33:44
wlan-client
```

Hat man ein WLAN-Gerät, das den 80211n-Modus unterstützt, muß die `hostapd.conf` angepasst werden. Zunächst werden die Fähigkeiten des Gerätes ausgelesen:

```
root@debian:~$ iw list
Wiphy phy0
    Band 1:
        Capabilities: 0x172
                    HT20/HT40
                    Static SM Power Save
                    RX Greenfield
                    RX HT20 SGI
                    RX HT40 SGI
                    RX STBC 1-stream
```

```

Max AMSDU length: 3839 bytes
No DSSS/CCK HT40
Maximum RX AMPDU length 65535 bytes (exponent: 0x003)
Minimum RX AMPDU time spacing: 2 usec (0x04)
HT RX MCS rate indexes supported: 0-7, 32
TX unequal modulation not supported
HT TX Max spatial streams: 1
HT TX MCS rate indexes supported may differ

```

Mit diesem Wissen kann man die hostapd.conf ergänzen:

```

### Kanaleinstellung
# freq          HT40-          HT40+
# 2.4 GHz       5-13          1-9
# 5 GHz         40, 48, 56, 64  36, 44, 52, 60
channel=7
# HT-Fähigkeiten (High Throughput)
ht_capab=[GF][SMPS-STATIC][HT40+][HT40-][SHORT-GI-20][SHORT-GI-40][RX-STBC1][MAX-AMSDU-3839]
# WMM aktivieren (siehe man hostapd.conf)
wmm_enabled=1
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=0
wmm_ac_be_acm=0
# N-Draft einschalten
ieee80211n=1

```

Viele Budget-USB-Sticks der N-Klasse unterstützen nur den 2,4GHz-Bereich. Da der freundliche Access-Point zunächst nach anderen Stationen Ausschau hält, die möglicherweise auf dem gleichen Kanal funken, ist es bei hoher AP-Dichte möglich, das das Zuschalten des zweiten Kanals abgelehnt wird. Hier ist es von Vorteil das 5GHz-Band im Gepäck zu haben, während beispielsweise auf dem Lande ein Gerät mit nur 2,4GHz-Band ausreicht. Auf die Möglichkeiten dem Access-Point seine Höflichkeit auszutreiben, gehe ich hier nicht ein. Hier ein Beispiel in der Debug-Ausgabe von hostapd:

```

Scan for neighboring BSSes prior to enabling 40 MHz channel
Scan requested (ret=0) - scan timeout 10 seconds
40 MHz affected channel range: [2427,2477] MHz
Neighboring BSS: 00:23:1a:21:ff:dd freq=2412 pri=1 sec=5
40 MHz pri/sec mismatch with BSS 00:23:1a:21:ff:dd <2412,2432> (chan=1+) vs. <2442,2462>
20/40 MHz operation not permitted on channel pri=7 sec=11 based on overlapping BSSes

```

## Weblinks

- [http://wiki.ubuntuusers.de/WLAN\\_Router](http://wiki.ubuntuusers.de/WLAN_Router)
- <https://wiki.debian.org/de/Wireless>

## Quellenangaben

```

man dnsmasq
man wpa_supplicant
man wpa_supplicant.conf
man hostapd
zless /usr/share/doc/hostapd/examples/hostapd.conf.gz

```

Von „[http://wiki.debianforum.de/wiki/?title=WLAN-Access-Point\\_mit\\_hostapd\\_und\\_USB-Stick&oldid=11735](http://wiki.debianforum.de/wiki/?title=WLAN-Access-Point_mit_hostapd_und_USB-Stick&oldid=11735)“

Kategorien: Netzwerk | Mobile Netzwerke | Getestet

### Meine Werkzeuge