

Connecting to a Wireless Network Using WPA/WPA2 for Authentication and Encryption#

Core Networking supports connecting to a wireless network using the more secure option of WPA (WiFi Protected Access) or WPA2 (802.11i) protocols. The application which manages this feature is called *wpa_supplicant*. The *wpa_supplicant* application can manage your connection to a single access point, or can manage a configuration which includes settings for connections to multiple wireless networks (SSID) either implementing WPA, or WEP to support roaming from network to network. *wpa_supplicant* supports IEEE802.1X EAP Authentication (referred to as WPA), WPA-PSK, and WPA-NONE (for ad-hoc networks) key management protocols along with encryption support for TKIP and AES (CCMP). A WAP for a simple home or small office wireless network would likely use WPA-PSK for the key management protocol, while a large office network would use WAP along with a central authentication server such as RADIUS.

To enable a wireless client (or supplicant) to connect to a WAP configured to use WPA, you must first determine the network name (as described above) and the authentication and encryption methods used from your network administrator. *wpa_supplicant* uses a configuration file (*/etc/wpa_supplicant.conf* by default) to configure its settings and then runs as a daemon in the background. You can also use the utility *wpa_cli* to change the *wpa_supplicant* configuration while it is running. Changes done by the *wpa_cli* utility will be saved to the */etc/wpa_supplicant.conf* file.

The */etc/wpa_supplicant.conf* file has a rich set of options which can be configured, but *wpa_supplicant* also makes use of various default settings which help simplify your wireless configuration.

If you are connecting to a WAP and your WPA configuration consists of a network name (SSID) and a pre-shared key, your */etc/wpa_supplicant.conf* (*make sure that this file is only readable and writable by root since it contains the key information in clear text*) would look like this:

```
network={
  ssid="my_network_name"    #The name of the network you wish to join
  psk="1234567890"         #The preshared key applied by the access point
}
```

Start *wpa_supplicant* as:

- `wpa_supplicant -B -i ral0 -c /etc/wpa_supplicant.conf`

Where *-i* specifies the network interface, and *-B* causes the application to move to the background.

wpa_supplicant will by default negotiate the use of the WPA protocol, WPA-PSK for key-management and TKIP or AES for encryption. Infrastructure mode is used by default.

Once the interface status is active (`ifconfig ral0`, where *ath0* is the interface name), you can apply the appropriate TCP/IP configuration (See TCP/IP Configuration in a Wireless Network).

If you were to create an ad-hoc network using WPA, your */etc/wpa_supplicant.conf* file (*make sure that this file is only readable and writable by root since it contains the key information in clear text*) would look like this:

```
network={
  mode=1                    #This sets the mode to be ad-hoc, 0 represents Infrastructure mode
  ssid="my_network_name"    #The name of the ad-hoc network
  key_mgmt=NONE             #Sets WPA-NONE
  group=CCMP                #Use AES encryption
  psk="1234567890"         #The preshared key applied by the access point
}
```

Start wpa_supplicant with:

- wpa_supplicant -B -i ath0 -c /etc/wpa_supplicant.conf

Where -i specifies the network interface, and -B causes the application to move to the background.

Personal-level Authentication and Enterprise-level Authentication#

WPA is designed to have two authentication methods. One is WPA-Personal / WPA2-Personal, which uses the pre-shared key which is the same passphrase shared by all network users. The other is WPA-Enterprise / WPA2-Enterprise, which uses an 802.1X authentication server which is RADIUS- based to authenticate each user. This section deals with the Enterprise-level authentication.

There are five Enterprise-level authentication methods that have been selected for use within the WiFi certification body. They are (1) EAP-TLS, which is the initially certified method. Both the server's certificates and the user's certificates are needed. (2) EAP-TTLS/MSCHAPv2: TTLS is short for "Tunnelled TLS". It works by first authenticating the server to the user via its CA certificate. The server and the user then establish a secure connection (the "tunnel"), and through the secure tunnel, the user gets authenticated. There are many ways of authenticating the user through the tunnel. The EAP-TTLS/MSCHAPv2 uses MSCHAPv2 for this authentication. (3) PEAP/MSCHAPv2: PEAP is the second most widely supported EAP after EAP-TLS. It is similar to EAP-TTLS, however, it requires only a server-side CA certificate to create a secure tunnel to protect the user authentication. Again, there are many ways of authenticating the user through the tunnel. The PEAP/MSCHAPv2 again uses MSCHAPv2 for authentication. (4) PEAP/GTC: This uses GTC as the authentication method through the PEAP tunnel. (5) EAP-SIM: This is for the GSM mobile telecom industry.

On io-pkt, we support all above except the EAP-SIM. Certificates would be placed in /etc/cert/user.pem and CA certificates in /etc/cert/root.pem. The following example is the network definition for wpa_supplicant for each of the above Enterprise level authentication.

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
```

```
# 3.1.2 linksys -- WEP
```

```
network={
    ssid="linksys"
    key_mgmt=NONE
    wep_key0="LINKSYSWEPKEY"
}
```

```
# 3.1.3 linksys -- WPA
```

```
network={
    ssid="linksys"
    key_mgmt=WPA-PSK
    psk="LINKSYSWPAKEY"
}
```

```
# 3.1.4 linksys -- WPA2
```

```
network={
    ssid="linksys"
    proto=RSN
    key_mgmt=WPA-PSK
    psk="LINKSYS_RSN_KEY"
}
```

```
# 3.1.5.1 linksys -- EAP-TLS
```

```
network={  
  ssid="linksys"  
  key_mgmt=WPA-EAP  
  eap=TLS  
  identity="client1"  
  ca_cert="/etc/cert/root.pem"  
  client_cert="/etc/cert/client1.pem"  
  private_key="/etc/cert/client1.pem"  
  private_key_passwd="wzhang"  
}
```

```
# 3.1.5.2 linksys -- PEAPv1/EAP-GTC
```

```
network={  
  ssid="linksys"  
  key_mgmt=WPA-EAP  
  eap=PEAP  
  identity="client1"  
  password="wzhang"  
  ca_cert="/etc/cert/root.pem"  
  phase1="peaplabel=0"  
  phase2="auth=PEAPv1"  
}
```

```
# 3.1.5.3 linksys -- EAP-TTLS/MSCHAPv2
```

```
network={  
  ssid="linksys"  
  key_mgmt=WPA-EAP  
  eap=TTLS  
  identity="client1"  
  password="wzhang"  
  ca_cert="/etc/cert/root.pem"  
  phase2="auth=MSCHAPv2"  
}
```

```
# 3.1.5.4 linksys -- PEAPv1/EAP-MSCHAPV2
```

```
network={  
  ssid="linksys"  
  key_mgmt=WPA-EAP  
  eap=PEAP  
  identity="client1"  
  password="wzhang"  
  ca_cert="/etc/cert/root.pem"  
  phase1="peaplabel=0"  
  phase2="auth=MSCHAPV2"  
}
```

Run

```
wpa_supplicant -i if_name -c full_path_to_your_config_file
```

to pick up the configuration file and get the supplicant to perform the required authentication to get access to the ~Wi-Fi network.