

百度一下

您查询的关键词是：**wpa\_supplicant.conf** 详解 以下是该网页在北京时间 2017年01月24日 21:57:40 的快照；

如果打开速度慢，可以尝试[快速版](#)；如果想更新或删除快照，可以[投诉快照](#)。

百度和网页 [http://blog.csdn.net/qg\\_22716879/article/details/51416322](http://blog.csdn.net/qg_22716879/article/details/51416322) 的作者无关，不对其内容负责。百度快照谨为网络故障时之索引，不代表被搜索网站的即时页面。

## [xuin 的博客](#)

保持足够的精力来应对万变的世界

-  [目录视图](#)
-  [摘要视图](#)
-  [订阅](#)

[CSDN学院招募微信小程序讲师啦](#)


[程序员简历优化指南！](#)


[【观点】移动原生App开发 PK HTML 5开发](#)

[云端应用征文大赛，秀绝招，赢无人机！](#)

# wpa\_supplicant 的配置说明文件 wpa\_supplicant.conf 译文

2016-05-18 08:39 4662人阅读 [评论 \(0\)](#) [收藏](#) [举报](#)

 分类：

通信/网络 (3) 

[作者同类文章 X](#)

## wpa\_supplicant 配置文件的例子

- 这个配置文件描述的格式和列表都是可用的选项
- 请阅读“examples”子目录下简单的配置例子
- 空行和空字符以及以“#”开头的字符都会被忽略
- 注意：这个文件中可能包含密码信息并且在多用户系统中
- 只有root用户才生效
- 注意：为确保当wpa\_supplicant运行在后台时，允许修改工作目录。
- 在这个配置文件中的文件路径应该为绝对路径
- 是否允许wpa\_supplicant更新（覆盖）配置文件
- 这个选项允许修改配置后wpa\_supplicant可以覆盖配置文件
- （eg:通过wpa\_cli加入新的网络语句块、wpa\_gui写入配置、密码改变等情况）
- 这对于wpa\_cli和wpa\_gui能够永久地修改配置是必需的。
- 请注意允许更新配置时应将该处的注释“#”移除

#update\_config=1

- 全局配置（在所有网络语句块中共享）
- 控制接口的参数，如若参数是指定的，wpa\_supplicant将会为外部程序打开控制接口从而管理wpa\_supplicant。字符串的含义取决于使用何种接口机制，对于所有案例，存在与配置中的参数决定了控制接口是否使能
- 对于 UNIX的域名套接字（在linux和BSD默认使用）：这个目录是为了监听从外部程序（CLI/GUI, etc.）对于查看状态信息和配置内容的请求。
- 套接字文件的命令将会基于接口的名称，因此如果有多个接口可用，多个wpa\_supplicant程序可以运行在用同一个时刻。

- /var/run/wpa\_supplicant是默认的也是推荐使用的存放套接字的目录，wpa\_cli将会在尝试和wpa\_supplicant连接时使用它。
- 通过设置该目录从而只允许一组成员使用套接字，控制接口的接入控制可以得到配置。这样wpa\_supplicant就有可能以root权限运行（当需要改变网络配置时和打开原始套接字时），同时仍然允许GUI/CLI 部分运行在非root权限的用户下。然而，因为该控制接口可以改变网络配置，在很多情况下这个接口需要被保护。默认情况下，wpa\_supplicant配置为使用gid 0（组ID为0 即root用户）。如果你想允许非root权限的用户去使用该控制接口的话，加入新的组并且改变这个值匹配那个组。添加用户到改组是需要拥有控制接口。如果该变量被注释掉或则没有包含在配置文件中。当创建目录或套接字创建时，该组将不会更改默认值的值。
- 在配置目录和组时，使用如下格式：
  - #DIR=/var/run/wpa\_supplicant GROUP=wheel
  - #DIR=/var/run/wpa\_supplicant GROUP=0
- (groud 可以是组的名称或者gid)
- 对于UDP的连接（windown下默认为该协议连接）改值将会被忽略。
- 该变量只是用于选择控制接口被创建，该值可以设置为udp（ctr\_interface=udp）。
- 对于windows的命名管道，该值可以被用于设置控制接口的安全描述符。可以使用安全描述符字符串格式设置（可看<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/>），该描述符需要加上”SDDL= “的前缀。例如，ctr\_interface=SDDL=D 该语句将会设置一个空的DACL（拒绝所有连接）。可在文档“README-Windows.txt”获取更多关于SDDL字符串格式的信息。

ctrl\_interface=/var/run/wpa\_supplicant

- IEEE 802.1X/EAPOL 版本
- wpa\_supplicant 的执行是基于 IEEE 802.1X-2004定义的EAPOL 第二版本 的标准。然而很多接入没有正确地遵循这一新版本号(他们似乎完全放弃了这一架构)。为了能让wpa\_supplicant和和这些接入点进行交互操作，改版本号默认设置为1，以下的配置数字可以用于设置为新版本（2）。
- 注意：当使用MACsec，eapol\_version 应该设置为3，定义在IEEE std802.1X-2010中

eapol\_version=1

- 接入点的扫描和选择
- 默认情况下，wpa\_supplicant请求驱动程序执行接入点的扫描并使用扫描结果选择合适的接入点。另外一种选择是允许设备驱动兼顾接入点的扫描和选择并且使用wpa\_supplicant只是运行基于IEEE 802.11 的EAPOL(extensible Authentication protol over Lan 基于局域网的可扩展认证协议)联系设备驱动的信息。
- 参数：0：wpa\_supplicant启动扫描和接入点选择；如果和当前可找到可用的网络不匹配，一个新的网络（IBSS 或者AP 模式操作）可能会被启动（如果配置）（默认）
- 1：设备驱动兼顾扫描、接入点选择和IEEE 802.11 连接参数（eg:WPA IE 一代）；这个模式也可以被用于使用IEEE 802.1X 模式的 non-WPA 设备驱动；不要试图和接入点取得联系（i.e.，外部程序需要控制连接）。这种模式同样必须在使用有限以太网时使用。
- 注意：macsec\_qca 设备驱动是一个执行macsec 特征的以太网设备类型
- 2：和参数0类似，但是通过使用安全的规则和SSID（不能为BSSID）和接入点取得联系；这个参数可以用于，例如，使用增强型和NDIS(network driver interface standard)的设备驱动来操作隐藏了的SSID和优化漫游；在这种模式下，在配置文件中的网络语句块将会一个个尝试连接直到设备驱动报告取得联系成功；每一个网络语句块应该有相对于密码管理（key\_mgmt）、配对法（pairwise）、用户组（group）、协议变量（proto variables） 明确的安全规则（在列表中只能有一个安全规则选项）
- 注意：当ap\_scan=2 不应该被用于使用nl80211 设备驱动接口中（当前linux的接口）。ap\_scan=1 是工作在nl80211的优化工作。为了寻找隐藏了的网络，scan\_ssid=1可以用于nl80211 的网络语句块中。
- 当使用IBSS 或者 AP 模式时，ap\_scan=2模式可以在忽略扫描结果的情况下立刻建立一个新的网络。ap\_scan=1 模式第一步将会试图扫描周围存在的网络。只有在无可匹配的网络情况下才会创建一个新的IBSS或者AP模式的网络。

ap\_scan=1

- 对等网络管理器的位置
- 默认情况下，wpa\_supplicant为开放式的网络运行对等网络管理器（MPM）。然而，如此设备驱动可以执行MPM，为了使用设备驱动版本号，你可能需要将它设置为0。当AMPE(automated message processing exchange 自动信息交换程序)使能时，wpa\_supplicant的MPM将会总是有用的。
- 0：MPM在设备驱动中被执行
- 1：wpa\_supplicant 提供一个处理对等（默认）的MPM

user\_mpm=1

- 最大的匹配链路数量（0-255；默认为99）
- 当前网络维持的最大网络匹配数量

max\_peer\_link=99

- 检测STA静止的超时时间（秒）（默认为300秒）
- 这个超时时间是用于STA网络清理不活动的站点。

```
mesh_max_inactivity=300
```

- cert\_in\_cb - 时候在事件跳转过程中包含匹配证书。
- 这个选项控制是否在EAP匹配认证事件中包含服务器认证和认证链节的匹配认证。在默认情况下该选项是激活的。

```
cert_in_cb=1
```

- EAP(extensible Authentication protocol)快速重认证
- 默认情况下，快速重认证对于支持所有EAP模式的情况下是激活的。这个参数可以被用于取消快速重认证。一般情况下，没有必要取消它。

```
fast_reauth=1
```

- OpenSSL(安全套接字库)工具支持
- 这些选项可以被用于加载OpenSSL工具
- 目前支持的两个工具如下所示：
- 它们都是来自opensc 工程 (<http://www.opensc.org/>)
- 默认情况下没有工具被加载。

```
//激活opensc工具
opensc_engine_path=/usr/lib/opensc/engine_opensc.so
//激活pkcs11工具
pkcs11_engine_path=/usr/lib/opensc/engine_pkcs11.so
//通过 pkcs11工具配置pkcs11模块请求路径
pkcs11_module_path=/usr/lib/pkcs11/opensc-pkcs11.so
```

- OpenSSL 密码字符串
- 这是配置OpenSSL默认密码的特定配置选项。如果没有设置，将会使用默认的“DEFAULT:!EXP:!LOW”。查看<https://www.openssl.org/docs/apps/ciphers.html> OpenSSL关于密码配置套件的文档。只有在wpa\_supplicant在建立是使用OpenSSL时才有效。

```
openssl_chiphers=DEFAULT:!EXP:!LOW
```

- 动态EAP方式
- 如果EAP建立动态共享对象文件，它们需要在使用网络语句块之前 在这里被加载。默认情况下，EAP方式在建立时是静态包含的，所以，这些选项是不必要的。

```
load_dynamic_eap=/usr/lib/wpa_supplicant/eap_tls.so
load_dynamic_eap=/usr/lib/wpa_supplicant/eap_md5.so
```

- 设备驱动程序的接口参数
- 这个区域可以被用于配置随意的设备驱动接口参数。选择设备驱动接口的格式是指定的。在很多案例中这个区域是不起作用的。

```
driver_param="field=value"
```

- 国家语言代码
- 当前设备驱动操作的国家语言代码

```
country=US
```

```
//PMKSA最大的存活时间（秒）；默认为43200
dot11RSNAConfigPMKLifetime=43200
//重认证的阈值（PMK存活时间的百分比）；默认为70
dot11RSNAConfigPMKReauthThreshold=70
//安全关联协议的超时时间（秒）；默认为60
dot11RSNAConfigSATimeout=60
```

- 建立Wi-Fi保护参数
- 设备通用唯一的标识符（UUID；参照RFC 4122）
- 如果没有配置，UUID将会被设置为本地MAC地址

```
uuid=12345678-91bc-def0-1234-56789abcdef0
```

```
//设备名称
//对设备使用友好的描述；UTF-8编码的32个字节
device_name=Wireless Client
//制造商
//设备的制造商（64个ASCII特征码）
```

```

manufacturer=Company
//型号
//设备型号（32个ASCII特征码）
model_name=cmodel
//型号编号
//额外的设备描述（32个ASCII特征码）
model_number=123
//序列号
//设备的序列号（32个ASCII特征码）
serial_number=12345

```

- 设备的主要类型
- 使用：--的格式
- categ：整型类型的数值
- OUI =组织唯一标识符；四个八位字节的16进制编码数值；WPS OUT默认为0050F204
- subcateg =OUI-特定的整型类型数值子类
- 例如：
- # 1-0050F204-1 (Computer / PC)
- # 1-0050F204-2 (Computer / Server)
- # 5-0050F204-1 (Storage / NAS)
- # 6-0050F204-1 (Network Infrastructure / AP)

```
device_type=1-0050F204-1
```

- 操作系统版本
- 操作系统版本数字的四个八位字节（16进制字符串）

```
os_version=01020300
```

- 配置方法
- 列表为所支持的配置方法
- 可行的方法：usba 以太网显示 ext\_nfc\_token int\_nfc\_token nfc\_interface push\_button keypad virtual\_display physical\_display virtual\_push\_button physical\_push\_button 标签

```

//对于 WSC 1.0
//配置方法=标签、显示、按钮、键盘
config_methods=label display push_button keypad
//对于 WSC 2.0
//配置方法=标签、虚拟显示、虚拟按钮、键盘
config_methods=label virtual_display virtual_push_button keypad

```

- 认证进程
- 0=进程内部接受凭证（默认）
- 1=进程不接受凭证；只是通过在ctrl\_iface上到外部程序
- 2=进程在内部接受凭证；只是通过在ctrl\_iface上到外部程序

```
wps_cred_processing=0
```

- WPS M1 的属性，比如window7垂直兼容
- 属性内容被添加到M1中（16位进制字符串）

```
wps_vendor_ext_m1=000137100100020001
```

- WPS的令牌-NFC密码
- 这些参数可以被用于配置一个固定的站点NFC密码。这个可以通过某种方式产生，例如使用nfc\_pw\_token 。当这些参数被使用时，该站点假设已经包含匹配NFC密码部署。（例如：编写记录在nfc\_pw\_token中的NDEF）

```

wps_nfc_dev_pw_id: Device Password ID (16..65535)
wps_nfc_dh_pubkey: Hexdump of DH Public Key
wps_nfc_dh_privkey: Hexdump of DH Private Key
wps_nfc_dev_pw: Hexdump of Device Password

```

- 通过添加WPS优先使用网络
- 这个优先值将会被设置到每个网络配置文件（执行WPS 协议时添加的）

```
wps_priority=0
```

- 保存在记忆体中最大的BSS 实体数量
- 默认值为：200
- 这个选项可以被设置限制BSS 实体（扫描结果缓存）的内存使用。在使用ap\_scan=1模式时在环境中可能需要一个比较大的值存储大量的接入点

```
bss_max_count=200
```

- 自动扫描
- 这是一个设置自动扫描以下格式的接口的配置参数
- autoscan=<自动扫描模块名称>:<模块参数>
- 自动扫描就像没有连接或者不活动的bgscan
- 例如指数模型上的参数将会为:
- autoscan=exponential:3:300
- 以上选项表示每次扫描的延时是基于3的指数的延时时间
- 最大为300s (3, 9, 27, ... 300).
- 对于周期性的模型, 参数将会是<固定的整数>
- #autoscan=periodic:30
- 以上参数表示30S一个周期自动扫描一次
- SSID过滤-基于SSID扫描结果过滤
- 0 表示对扫描结果不过滤 (默认)
- 1 表示从扫描结果中或者扫描结果缓存表中过滤出配置文件中的SSID

```
filter_ssids=0
```

- 外部存储后端的密码 (密码口令, 等等)
- 格式: <后端名称>[:<后端参数选项>]

```
ext_password_backend=test:pw1=password|pw2=testing
```

- 取消P2P功能

```
p2p_disabled=1
```

- 移除不活动的STA的超时时间 (默认为300S)
- 该超时时间值用于在P2P GO 模式下清除不活动的站点

```
p2p_go_max_inactivity=300
```

- P2P GO 密码长度 (8-63)
- 这个参数控制在GO 生成的随机密码的长度。(默认为8)

```
p2p_passphrase_len=8
```

- 并行进行P2P迭代搜索的额外延时时间
- 这个值增加额外的延时 (ms) 让p2p\_find友好地并行操作, 从而避免无线点占用100%的资源。该值默认为500ms

```
p2p_search_delay=500
```

- 机会性密钥缓存 (已称为主动型密钥缓存) (默认设置)
- 这个参数可以用于设置默认的proactive\_key\_caching参数行为。默认情况下, OKC是无效的除非使能全局变量okc=1 或者每个网络中proactive\_key\_caching=1。如若okc=1, OKC默认是有效的, 不允许每个网络中的proactive\_key\_caching=0

```
okc=0
```

- 默认的保护管理架构
- 这个参数可以用于设置ieee80211w 参数的默认行为。默认情况下, PMF无效除非全局变量pmf=1/2, 或者每个网络中都配置ieee80211w=1/2 这个参数。当pmf=1/2 时PMF 默认是有效的和可被请求的。在每个网络中也可以使用ieee80211w 参数设置为失效。

```
pmf=0
```

- 启用SAE有限循环群优先顺序
- 默认情况下 (该参数没有设置) 组19 (组 ECC定义在超过256位的基本指令领域) 是首选的选项, 其他组也是激活的。如果该参数被设置, 改组将会尝试控制命令。改组的值在[IANA注册表](#)中列出

```
sae_groups=21 20 19 26 25
```

```
//DTIM周期的默认值 (如果没有在网络语句块中被重写)
```

```
dtim_period=2
```

```
//信标间隔的默认值 (如果没有在网络语句块中被重写)
```

```
beacon_int=100
```

- 信标和探测响应帧的基本属性
- 该参数可以被用于增加额外的基本属性到信标和探测响应帧的末尾。这些属性的格式是一个十六位进制转储的原始信息元素（一个或多个元素的 id+len+payload），这用于 AP 和 P2P GO 模式下。

ap\_vendor\_elements=dd0411223301

- 忽略请求扫描结果命令
- 当我们触发扫描命令时，设备驱动可能有扫描结果的返回信息，这个参数可以用于配置类似于被忽略旧的信息而不是更新内部BSS 表格。

ignore\_old\_scan\_res=0

- scan\_cur\_freq：是否只在当前频率下扫描
- 0：在所有可用的频率下扫描（默认）
- 1：如果另一个同一个无线 VIF已经取得关联下，在当前操作频率下扫描
- 默认的MAC 地址规则
- 0：使用永久的MAC地址
- 1：给每个连接的ESS使用随机的MAC地址
- 2：类似于1，但是保持 OUI（本地管理位集）
- 默认情况下，使用永久的MAC地址除非通过每一个网络mac\_addr参数改变规则。全局设置mac\_addr=1 可以用于改变这个默认的配置行为。

mac\_addr=0

- 随机MAC地址的存活时间（秒）（默认为60s）

rand\_addr\_lifetime=60

- 每个关联操作的MAC地址规则（scanning(扫描)、ANQP（接入网络查询协议））
- 0=使用永久MAC地址
- 1=使用随机的MAC地址
- 2=类似于1，但是使用保持的MAC 地址（本地管理位集）

preassoc\_mac\_addr=0

- Interworking (IEEE 802.11u)
- 激活Interworking

interworking=1

- 纯质的ESS标志符
- 如果这个被设置，扫描将会用于回应只属于特定的Homogeneous ESS的请求，这个只用于interworking激活的情形。

hessid=00:11:22:33:44:55

- 自动选择网络的行为
- 0=不自动通过互联网选择
- 1=执行互联网网络选择当一个或者多个认证已经配置和没有扫描到匹配的网络语句块

auto\_interworking=0

- 认证块
- 用于自动网络选择的每个认证将会构造为一组参数，当使用interworking\_select and interworking\_connect命令时，将会和APs发布的信息比较
- 凭证字段：
- 暂时的(temporary)：认证是否是暂时的并且不被保存
- 优先级(priority)：优先级组
- 默认情况下，所有的网络和认证都将获取同等的优先级组（0）这个字段可以被用于认证的更高优先级（和网络语句块中的结构体wpa\_ssid很类似），这样可以改变自动选择网络的行为。拥有更高优先级的匹配的网络（要么基于激活的网络语句块要么是认证凭证）将会被选择。
- psc:使用PC / SC和SIM/ USIM卡
- realm(领域):家庭领域的互联网
- username(用户名): 互联网网络选择的用户名
- password(密码): 互联网网络选择的密码
- ca\_cert: 互联网选择的CA认证



- `client_cert`: 客户端认证文件的路径 (PEM/DER)
- 以上字段适用于在客户端用认证/密钥来认证 (EAP-TLS) 的网络自动选择。文件的路径应该为绝对路径 (`wpa_supplicant` 运行于后台时可能会改变工作路径)。另外一个blob的配置可以将此设置为blob: `//blob_name`
- 私人密钥: 私人密钥 (PEM/DER/PFX) 的文件路径  
当使用PKCS#12/PFX文件时 (`.p12/.pfx`), `client_cert` 应该被注释掉。这种情况下私人密钥和认证凭证将会从PKCS#12文件中读出。完整的路径需要为绝对工作路径 (`wpa_supplicant` 运行于后台时可能会改变工作路径)

通过从客户端认证中退出, windows下的认证储存可以被使用, 并且私人密钥可以通过以下格式配置:

`cert://substring_to_match`

`hash://certificate_thumbprint_in_hex`

举个例子: `private_key="hash://63093aa9c47f56ae88334c7b65a4"`

注意当将`wpa_supplicant`作为应用运行时, 用户认证储存区 (自己的用户账户) 将被使用。而当运行`wpa_supplicant`作为服务端时计算机储存区 (计算机账户) 将被使用。

另外, 一个blob的配置可以将此设置为blob: `//blob_name`

- `private_key_passwd`: 私人密钥的密码文件
- `imsi`: IMSI in | | '-' | format
- `milennage`: Milennage parameters for SIM/USIM simulator in :: format
- `domain`: Home service provider国内服务提供商 FQDN(s)  
以上选项将被用于比较域服务商名称列表从而计算出是否该AP是在家用SP下操作。多域接入点可以被用于配置被认为家庭网络的非传统FQDNS
- `roaming_consortium`: (漫游联合体) Roaming Consortium OI  
如果`roaming_consortium_len`非空, 这个字段包含可以决定哪个接入点支持凭证认证的漫游联合体。这是非传统的领域参数的用法。当使用漫游联合体匹配网络时, 因为NAI域的信息可能不可用或者牵强, EAP参数需要使用凭证预配置。
- `eap`: EAP预编译的方法  
这个字段可以用来指定哪个EAP方法使用这个凭证。如若没有设置, EAP方法设置为自动基于ANQP信息 (eg: NAI Realm) .
- `phase1`: 预编译阶段1 (外部认证) 参数, 这个字段设置和 'eap' 参数类似
- `phase2`: 预编译阶段2 (内部认证) 参数, 这个字段设置和 'eap' 参数类似
- `excluded_ssid`: 过滤SSID  
这个选项字段可以用于从配置网络中过滤指定的SSID(S), 多选项可以增加更多SSID
- `roaming_partner`: 漫游合作伙伴的信息  
这个字段选项可以用于配置和漫游伙伴的首选选项。该配置字段使用以下格式:  
,<0/1 exact match>,,<\* or country code>  
non-exact 表示任何子域都可以匹配该接口; priority是设置更高优先级, 范围在0-255
- `update_identifier`: 更新标识符 (PPS MO ID) (Hotspot 2.0 PerProviderSubscription/UpdateIdentifier)
- `provisioning_sp`: 服务提供商FQDN提供的凭证  
这个字段选项可以用于跟踪服务提供商SP提供的凭证并且寻找PPS MO (./Wi-Fi/).
- 最大的回程链路的阈值 (PPS/

`min_dl_bandwidth_home`

`min_ul_bandwidth_home`

`min_dl_bandwidth_roaming`

`min_ul_bandwidth_roaming`

- `max_bss_load`: BBS (扫描结果缓存表) 的最大通道负荷 (1-255)  
(PPS/

`req_conn_capab=6, 22, 80, 443`

For example, IPSec/IKE:

`req_conn_capab=17:500`

`req_conn_capab=50`

- `oscp`: 是否使用/请求OCSP检查服务凭证  
0=没有使用OCSP (扩展的认证状态 TLS)  
1=试图使用OCSP, 但不请求应答  
2=请求接受有效的OCSP应答

- sim\_num: 使用多SIM 卡设备时SIM卡的标识符

下面对语句块举例子:

```
cred={
    realm="example.com"
    username="user@example.com"
    password="password"
    ca_cert="/etc/wpa_supplicant/ca.pem"
    domain="example.com"
}

cred={
    imsi="310026-000000000"
    milenage="90dca4eda45b53cf0f12d7c9c3bc6a89:cb9cccc4b9258e6dca4760379fb82"
}

cred={
    realm="example.com"
    username="user"
    password="password"
    ca_cert="/etc/wpa_supplicant/ca.pem"
    domain="example.com"
    roaming_consortium=223344
    eap=TLS
    phase2="auth=MSCHAPV2"
}
```

- Hotspot 2.0 (热点2.0)  
hs20=1

## 网络语句块

- 在配置文件中, 每个网络 (通常是AP 的共享SSID) 被视为独立的模块来配置
- 该网络语句块是有优先配置顺序的 (第一次匹配)。

-网络语句块字段:

- disabled:  
0=该网络可以被使用 (默认)  
1=该网络语句块是失效的 (可以通过ctrl\_iface激活 eg:使用wpa\_cli或者wpa\_gui)
- id\_str: 外部脚本的网络标识字符。通过wpa\_cli外部执行脚本将WPA\_ID\_STR作为环境变量, 使得配置特定网络变得更加容易。
- ssid:SSID(必须的); 一种如下格式的网络名称:  
引用双精度的ASCII字符  
一个十六进制的字符串 (SSID每个字节的两个字符)  
打印的字符: 字符串
- scan\_ssid:  
0=不扫描这个通过特定的探测请求帧得到的SSID (默认)  
1=扫描通过特定探测请求帧得到的SSID (这个可以用于寻找不接受广播的APs或者使用多SSIDs ;这个将会增加扫描延长时间, 所以在有必要时才激活此选项)
- bssid:BSSID(不必要的、可选择的); 如果设置了的话, 这个网络语句块将被用于使用BSSID配置文件和APs取得关联。
- priority: 优先级组 (整数)
- 默认情况下, 所有网络将会获得相同的优先级组 (0), 如果有一些更加可取的网路, 这个字段可以被用于改变wpa\_supplicant在BSS中选择网络时的顺序。这个优先级组将会迭代减少优先级 (i.e, 优先级数值越大, 就越早在扫描结果中得到匹配), 对于相同的优先级组, 将会根据安全规则、信号强度等来作为选择的依据。
- 请注意在scan\_ssid=1 和 ap\_scan=2 模式下AP的扫描不能够使用这个优先级组去选择扫描顺序。相反的他们尝试使用配置文件中的网络顺序。
- mode: IEEE 802.11操作模式
- 0 = 基础架构模式 (管理模式), i.e, 和AP取得关联 (默认)
- 1 = IBSS (自组网、点对点)
- 2 = AP (接入点)
- 注意: IBSS只能在key\_mgmt NONE (明码文本和静态WEP) 和WPA-PSK(proto = RSN)下使用。另外, key\_mgmt=WPA-NONE (固定的TKIP/CCMP密码组) 也是向后兼容的, 但是这种用法是过时的。WPA-None 需要以下的网络语句块选项:
- proto=WPA, key\_mgmt=WPA-NONE, pairwise=NONE, group=TKIP (or CCMP, 两个不能同时存在), 并且密码也必须得到设置。
- frequency: IBSS的通道频率 (MHZ), 例如: 2412=IEEE 802.11b/g 是通道一的频率。这个值被用于配置IBSS(adhoc自组网)的初始化通道。在基础架构模式下它将会被忽略。另外, 这个值只用于创建IBSS站点。如果一个配置了SSID的IBSS网络已经存在, 其网络的频率值将会被用于替代这里的配置数值。
- scan\_freq: 扫描频率的列表



- 当扫描BSS时使用空间分离的频率表（MHZ）扫描。如果该网络的通道设置是已知的，该选项可以被用于优化扫描行为，避免扫描网络没有用到的通道
- Example: scan\_freq=2412 2437 2462
- freq\_list: 获得许可频率的数组
- 允许选择BSS的空间分离的频率表。如果设置了的话，当选择一个BSS时，扫描结果将考虑匹配任何特定的频率的网络。
- 这个选项也可以被设置到网络语句块外面。在这种情形下，它将限制特定的频率被扫描。
- bgscan: 后台扫描
- 通过设置为bgscan模块，wpa\_supplicant可以被配置后台扫描行为。为了漫游在ESS下，这些模块要求在后台运行扫描（i.e., 即在所有AP的单个网络语句块中使用相同的SSID）。这个bgscan参数可以使用以下格式：“:”
- 以下bgscan模块是可用的:
- 简单型: 根据信号强度周期性地后台扫描
- bgscan="simple:::"
- bgscan="simple:30:-45:300"
- 学习型: 通过网络学习使用的通道，避免bgscan扫描其他通道（实验型）。
- bgscan="learn::[:]"
- bgscan="learn:30:-45:300:/etc/wpa\_supplicant/network1.bgscan"
- 另外可通过以下命令取消bgscan:
- bgscan=""
- 这个选项也可以全局设置，应用于没有指定特定的bgscan参数。

proto: 公认的协议列表

- WPA=WPA/IEEE 802.11i/D3.0
- RSN = WPA2/IEEE 802.11i (可以使用WPA2作为RSN的别名)
- 如果没有设置将会默认使用WPA RSN

key\_mgmt: 公认认证密钥管理协议列表

- WPA-PSK = WPA 预共享密钥（这需要‘psk’字段）
- WPA-EAP = WPA 使用 EAP 认证
- IEEE8021X = IEEE 802.1X 使用EAP 认证并动态生成密钥
- NONE = 没有使用 WPA ; 可以使用明码密钥或者静态WEP
- WPA-PSK-SHA256 = 类似 WPA-PSK 但是使用增强型的SHA256-based加密算法
- WPA-EAP-SHA256 = 类似 WPA-EAP 但是使用增强型的SHA256-based加密算法
- 如果没有设置，将会模式设置为 WPA-PSK WPA-EAP

ieee80211w: 是否激活包含管理机制

- 0 = 取消（默认设置，除非通过和全局pmf参数改变）
- 1 = 可选择的
- 2 = 必要的
- 更多通用的配置选项基于PMF(包含管理机制protected management frames)认证程序:
- PMF enabled: ieee80211w=1 and key\_mgmt=WPA-EAP WPA-EAP-SHA256
- PMF required: ieee80211w=2 and key\_mgmt=WPA-EAP-SHA256
- （当使用WPA2-Personal时WPA-PSK and WPA-PSK-SHA256也是类似配置）

auth\_alg: 允许的IEEE 802.11 认证算法列表

- OPEN = 开放性系统认证（WPA/WPA2的必要选项）
- SHARED = 共享密钥认证（静态WEP密码的必要选项）
- LEAP = LEAP/Network EAP（只在LEAP中使用）
- 如果没有设置，将使用自动选择（如果LEAP允许作为一种EAP方法，LEAP开放系统将会被激活）

pairwise : WPA点对点（单播）支持的密码表格

- CCMP = AES 的CBC-MAC计数模式 [RFC 3610, IEEE 802.11i/D7.0]
- TKIP = 临时密钥完整性协议 [IEEE 802.11i/D7.0]
- NONE = 只使用组密码（不赞成使用，当APs 支持成对密钥是将被包含）
- 如果没有设置，将默认设置为: CCMP TKIP

group: 组（广播/组播）支持的密码列表

- CCMP = AES 的CBC-MAC计数模式 [RFC 3610, IEEE 802.11i/D7.0]
- TKIP = 临时密钥完整性协议 [IEEE 802.11i/D7.0]

- WEP104 = 104位 WEP(Wired Equivalent Privacy有效对等密钥)
- WEP40 = 40位 WEP(Wired Equivalent Privacy有效对等密钥)[IEEE 802.11]
- 如果没有设置，将会默认设置为： CCMP TKIP WEP104 WEP40

psk: WPA预共享密码; 256位预共享密码

- 在WPA-PSK模式下的密码可以使用64进制数字输入，即，32字节或者一个ASCII密码（在这种情况下，真正的PSK将会使用密码和SSID生成）。ASCII密码必须在8到63个字节之间（包含8字节和63字节）。拓展：拓展字段格式可以被用于表示PSK/passphrase储存在外部存储器中。
- 如果使用WPA-EAP这个字段不是必须的。
- 注意：其他一些工具，wpa\_supplicant可以被用于从ASCII 密码中生成256位密钥。这些进程将会占用很多CPU资源，并且在密码和SSID确实被改变的情况下再生成密钥可以优化wpa\_supplicant的启动和重构时间。

mem\_only\_psk: 是否只在内存中保持PSK/passphrase

- 0 = 允许psk/passphrase储存在配置文件中
- 1 = psk/passphrase不储存在配置文件中
- mem\_only\_psk=0

\* eapol\_flags: IEEE 802.1X/EAPOL 选项(bit field位字段)\*

- 在non-WPA模式下的动态WEP 密码
- bit0 (1): 要求动态生成单播的WEP密钥
- bit1 (2): 要求动态生成广播WEP密钥
- (3=要求两种类型的WEP密钥; 默认设置)
- 注意：当使用无线认证时（包括macsec\_qca设备驱动），为了能够完整地认证，需要将eapol\_flags设置为0

macsec\_policy: IEEE 802.1X/MACsec options选项

- 这决定了MACsec使用何种回话担保。这个目前只用于使用macsec\_qca 设备驱动接口时。
- 0 = 不使用MACsec（默认）
- 1 = 激活MACsec-必须是安全的，接受密码服务器的通知来决定是否使用一个安全会话与否

mixed\_cell : 这个选项可以被用于配置是否为所谓的混杂单元，即一个同样的SSID 网络使用明码和加密密钥

- 0 = 取消（默认）
- 1 = 激活

proactive\_key\_caching:

- 激活/取消WPA2的PMKSA随机缓存
- 0 = 取消（默认设置，除非使用全局参数okc来改变）
- 1 = 激活

wep\_key0..3: 静态WEP密码（双精度ASCII引用，比如“abcd”；或者没有引用的十六进制数，比如“012345678”）

- wep\_tx\_keyidx: Default WEP key index (TX) (0..3)

peerkey: 是否允许直接连接的peerkey协议(IEEE 802.11e DLS)，这个只用于 RSN/WPA2

- 0 = 取消（默认）
- 1 = 激活
- peerkey=1

wpa\_ptk\_rekey: PTK最大的存活时间（秒），这个选项可以被用于配置执行PTK密钥更新来缓解因为TKIP缺陷而受到的攻击

以下字段值用于内部EAP实施。

- eap: 可接受的空间分离的EAP方法列表
- MD5 = EAP-MD5（不安全和不生成原密码。不能用于WPA；在使用方法EAP-PEAP or EAP-TTLS中的第二阶段使用）
- MSCHAPV2 = EAP-MSCHAPv2（不能单独使用WPA；在使用方法EAP-PEAP or EAP-TTLS中的第二阶段中使用）
- OTP = EAP-OTP（）（不能单独使用WPA；在使用方法EAP-PEAP or EAP-TTLS中的第二阶段中使用）
- GTC = EAP-GTC（不能单独使用WPA；在使用方法EAP-PEAP or EAP-TTLS中的第二阶段中使用）
- TLS = EAP-TLS（客户端和服务端的认证）
- PEAP = EAP-PEAP（EAP认证隧道）
- TTLS = EAP-TTLS（EAP认证隧道或者PAP/CHAP/MSCHAP/MSCHAPV2认证）
- 如果没有设置，所有编译的方法都被允许

## identity:EAP的字符标识符

- 这个字段同样被用于配置使用EAP-PSK/PAX/SAKE/GPSK的NAI

anonymous\_identity : EAP的匿名标识符字符串（被用于支持不同的标识符隧道的非加密EAP类型的标识符，比如，EAP-TTLS）。这个字段也可以被用于EAP-SIM/AKA/AKA' 保存匿名标识符

password: EAP的密码字符串。这个字段既可以包含纯文本密码字符串（使用ASCII或者十六进制字符串）也可以一个使用hash:<32 hex digits>格式的NtPasswordHash（16字节MD4 哈希密码）

- NtPasswordHash只能用于MSCHAPv2 或者 MSCHAP (EAP-MSCHAPv2, EAP-TTLS/MSCHAPv2, EAP-TTLS/MSCHAP, LEAP); EAP-PSK (128-bit PSK), EAP-PAX (128-bit PSK), 和EAP-SAKE (256-bit PSK) 也可以使用该字段得到配置。

- 对于EAP-GPSK这是一个正确的psk长度。拓展：格式可以被用于表示密码储存在外部存储器中。

ca\_cert:CA 的认证文件（PEM/DER）的路径

- 这个字段有一个或者多个的受信任的CA认证。如果ca\_cert 和 ca\_path都没有包含文件路径，认证服务将不会验证。这是不安全的，并且当使用EAP-TLS/TTLS/PEAP时一个受信任的CA认证总是被配置。完整的路径应该为绝对路径（wpa\_supplicant可能在后天运行时改变工作路径）。

- 另外，这个字段可以被用于只执行服务器认证的匹配任务（SHA-256 hash of the DER encoded X.509 certificate），在这种情况下，服务器认证中CA认证可能被忽略，并且只验证服务器认证。这个可以通过以下格式来配置：

hash://server/sha256/cert\_hash\_in\_hex

- 例如：” hash://server/sha256/5a1bc1296205e6fdb3979728efe3920798885c1c4590b5f90f43222d239ca6a”

- 在window操作系统下，受信任的CA认证可以通过设置cert\_store://从系统认证储存器中加载，例如：  
ca\_cert=” cert\_store://CA” or ca\_cert=” cert\_store://ROOT” . 注意当将wpa\_supplicant作为应用程序运行时，用户（当前账户）认证储存器将是可用的，当运行wpasvc作为服务器时，计算器（计算机账户）储存器将是可用的。

\* ca\_path : CA认证文件（PEM）的目录路径。\*

- 这个路径可能包含OpenSSL格式的多CA认证。这种常见的使用方法是指系统那些安装在/etc/ssl/certs路径下的可信任的CA认证列表，如果配置了的话，这些认证将会被添加到可信任CAs列表中， ca\_cert可能也包含在这种案例中，但它不是必须的。

## client\_cert : 客户端认证文件的路径（PEM/DER）

- 完整的路径应该为文件的绝对路径（因为wpa\_supplicant运行在后台时可能会改变工作路径）

- 另外，名为blob的配置也可以被用来设置blob://.

private\_key : 客户端私人密钥文件路径（PEM/DER/PFX）

- 当使用 PKCS#12/PFX 文件（.p12/.pfx）， client\_cert应该被注释掉。这种情况下私人密钥和认证将会从PKCS#12文件中读出。完整的路径应该为文件的绝对路径（因为wpa\_supplicant运行在后台时可能会改变工作路径）

- 当client\_cert退出时并且配置私人密钥如以下格式时，windows 认证储存器可以被使用

cert://substring\_to\_match

hash://certificate\_thumbprint\_in\_hex

- 例如 private\_key=” hash://63093aa9c47f56ae88334c7b65a4”

- 注意：当wpa\_supplicant作为应用程序运行时，用户（当前账户）认证储存器将是可用的，当运行wpasvc作为服务器时，计算器（计算机账户）储存器将是可用的。

- 另外，名为blob的配置也可以被用来设置blob://.

\* private\_key\_passwd : 私人密钥文件的密码（如果遗漏，将会在控制界面中询问）\*

\* dh\_file: DH/DSA参数文件的路径（PEM 格式）\*

- 这个选项配置设置一个短暂的DH密码交换参数的文件。在大多数情况下，默认的RSA认证将不会在这种情况下使用。然而，使用短暂的DH密码交换可能会建立RSA认证。另外，DSA 密码算法总是使用短暂性DH密钥，这个选项可以用于实现保密。如果这个文件使用了DAS参数格式，它将会自动转化为DH参数

subject\_match: 子类字符串是针对认证服务器证书来匹配

- 如果这个字符串设置了的话，服务器证书只有在这个实体上包含了这个字符串才的情况下才能被接受。

- 该实体字符串使用以下格式配置：

/C=US/ST=CA/L=San Francisco/CN=Test AS/emailAddress=as@example.com

- 注意：因为这是一个子类字符串匹配，对于可能的一种中文实体域名情况下将不能安全地进行后缀匹配。对于这种应用案例，应该使用domain\_suffix\_match 或者 domain\_match来替代。

altsubject\_match : 分号分隔符将会选择和选择服务器认证证书实体名称相匹配

- 如果该字符被设置的话，如果服务器认证包含了一个扩展的选择项目名称实体，该认证才能被接受。

- altSubjectName字符串的格式: TYPE:VALUE
- Example: EMAIL:server@example.com
- Example: DNS:server.example.com;DNS:server2.example.com
- 支持的格式为: EMAIL, DNS, URI

#### domain\_suffix\_match:域名服务器名称的规则

- 如果设置的话，在SubjectAltName dNSName的AAAServer认证请求FQDN后缀匹配。如果匹配了一个找到DNS名称，约束条件将会被满足。如果没有DNS名称的值存在，这个约束条件是使用相同的后缀匹配来匹配中文实体名。
- 这里的后缀匹配表示逐一从顶级域名开始并且domain\_suffix\_match的所有标签应该包含在认证证书中和一个标签比较主机/域名。除了必要的标签外，认证证书可能包含额外的字级标签。
- 例如，domain\_suffix\_match=example.com 将会匹配test.example.com，但它将不会匹配test-example.com。

#### domain\_match:域名服务器的约束条件

- 如果设置的话，FQDN将会被用做一个要求完全匹配服务器认证证书中SubjectAltName 域名元素。如果一个匹配的域名被找到，这个约束条件将被满足。如果不存在域名名称值，这个约束就是在SubjectName CN中文 同样的完全匹配来比较得匹配。这种行为就和domain\_suffix\_match类似，但是有一个完全匹配是必须的，即不允许子域名和通配符匹配。使用不区分大小的比较，因此“Example.com” 可以匹配 “example.com”，但是将不会匹配 “test.Example.com”。

phase1 : 阶段一（外部认证：即TLS隧道）参数（成对的字段字符，比如：“peapver=0” 或者“peapver=1 peaplabel=1”）

- ‘peapver’ 可以被用于强制使用哪个版本的PEAP（0或者1）。
- ‘peaplabel=1’ 可以被用于在使用PEAPv1或者更新版本时导出密码的期间强制使用新的标签 “client PEAP encryption”，一些扩展的PEAPv1的实施还使用旧的标签 “client EAP encryption”，现在wpa\_supplicant已经使用 “client EAP encryption” 作为默认的值。一些服务器，比如辐射器可能需要强制配置peaplabel=1 来和PEAPv1进行交互操作；查看文件eap\_testing.txt 得到更多相关信息。
- ‘peap\_outer\_success=0’ 可以被用于终止EAP-Success 隧道的PEAP认证。这在一些执行 draft-josefsson-pppext-eap-tls-eap-05.txt（比如Lucent NavisRadius v4.4.0 with PEAP in “IETF Draft 5” mode模式）RADIUS服务器上。
- include\_tls\_length=1可以被用于强制使wpa\_supplicant 在所有没有碎片化的TLS管理事件中包含TLS 管理长度字段。
- sim\_min\_num\_chal=3 可以被用于配置EAP-SIM请求三个口令（默认情况下，它只接受到2个或者3个）
- result\_ind=1可以被用于使能EAP-SIM和EAP-AKA来指示保护结果。
- ‘crypto\_binding’ 该选项可以被用于控制PEAPv0加密绑定行为：
  - 0 =不适用加密绑定（默认）
  - 1 =当服务器支持时使用加密绑定
  - 2 = 请求加密绑定
- EAP-WSC (WPS)使用以下选项的格式: pin= or pbc=1.
- 对于无线IEEE 802.1X加密认证，“allow\_canned\_success=1” 可以被用于配置允许EAP-Success (and EAP-Failure)不经认证步骤的模式。一些交换机当强制端口被授权/不被授权或者认证服务器遥不可及是的一种备用选项才使用这种配置序列。
- 默认情况下，wpa\_supplicant丢弃这种免收流氓设备的潜在攻击的保护机制，但这个选项可以被用于取消在服务器/认证器不需要被认证的保护案例。

phase2 : 阶段二（TLS隧道的内部认证）参数（string with field-value pairs, e.g., “auth=MSCHAPV2” for EAP-PEAP or “auth=MSCHAPV2 auth=MD5” for EAP-TTLS）

- “mschapv2\_retry=0” 可以被用于取消 MSCHAPv2 在认证失败后重试密码。

TLS-based 方法可以用以下参数来控制TLS行为（这个通常是下第一阶段的参数，但是也可以当EAP-TLS在内部隧道使用时被用在第二阶段）

- tls\_allow\_md5=1 允许MD5-based 证书签名（取决于TLS库，为了足够安全这个选项默认取消）
- tls\_disable\_time\_checks=1 忽略证书有效期（这个选项要求TLS库接受认证，即使当前它们是不能用的，即已过期或尚未生效；这个应该只作为测试目的使用）
- tls\_disable\_session\_ticket=1 取消TLS外部探测会话
- tls\_disable\_session\_ticket=0 允许使用TLS外部探测会话（注意：如果没有设置，对于EAP-TLS/PEAP/TTLS将会自动设置为1，作为一个来破坏服务器执行认证的工作区，除非EAP工作区使用参数eap\_workaround=0来取消。）。对于EAP-FAST，这个参数必须设置为0（或者不设置，使用自动配置默认值）
- tls\_disable\_tlsv1\_0=1 取消使用TLSv1.0
- tls\_disable\_tlsv1\_1=1 取消使用TLSv1.1（已经发布更新的TLS版本的AAA服务器的工作区）
- tls\_disable\_tlsv1\_2=1 取消使用TLSv1.2（已经发布更新的TLS版本的AAA服务器的工作区）

以下认证/私人密钥字段将在使用EAP-TTLS 或者 EAP-PEAP 时内部第二阶段认证使用

- `ca_cert2`: CA认证证书的文件路径。这个文件可以有一个或者多个受信任的CA认证证书。如果不包含`ca_cert2` 和 `ca_path2` 服务器认证将不会生效。这是不安全的, 受信任的CA认证证书必须总是被配置。
- `ca_path2`: CA认证证书的目录路径 (PEM)
- `client_cert2`: 客户端认证证书的文件路径
- `private_key2`: 客户端私人密钥文件路径
- `private_key2_passwd`: 私人密码文件
- `dh_file2`: DH/DSA参数文件路径 (PEM格式)
- `subject_match2`: 通过子类字符串匹配认证服务器的凭证证书。查看`subject_match`获取相关细节信息。
- `altsubject_match2`: 以分号分隔字符串匹配认证服务器的备用实体名。查看`altsubject_match`获取相关细节。
- `domain_suffix_match2`: 域名服务器的约束, 查看`domain_suffix_match`获取更多细节。
- `fragment_size`: EAP片段最大的字节长度 (默认为1398)。  
这个值限制使用支持片段的EAP方法的片段长度 (例如: EAP-TLS and EAP-PEAP)。这个值应该被设置为足够小的值来使EAP消息能够适用于使用EAPOL的MTU网络接口。在大多数情况下默认是已经是合适的值了。
- `ocsp`: 是否使用/请求OCSP来检测服务器认证证书
- 0 = 不使用OCSP (外部TLS 认证状态)
- 1 = 尝试使用OCSP 但不是必要的请求
- 2 = 请求可用的OCSP
- `openssl_ciphers`: OpenSSL指定的密码配置 (这个选项可以被用于重写全局参数`openssl_ciphers` 往回看)

### EAP-FAST 变量

- `pac_file`: PAC实体的文件路径。`wpa_supplicant`将需要能够在PAC被满足或者拒绝时创建这个文件并且写入更新到这个文件中。完整的文件路径必须是绝对文件路径 (因为`wpa_supplicant`作为后台运行程序时可能会改变工作路径)。另外, `blob`名字配置可以被用于设置这个参数到`blob`:
  - `phase1: fast_provisioning`可以用来激活对EAP-FAST认证证书 (PAC) 的在线配置:
    - 0 = 取消
    - 1 = 允许未经身份认证的配置
    - 2 = 允许经身份认证的配置
    - 3 = 允许未经身份认证和身份认证的配置
- `fast_max_pac_list_len`=选项可以被用于设置PAC实体储存在PAC列表的最大数量 (默认为10)
- `fast_pac_format=binary` 可以被用于选择为了保存一些空间 (默认使用约为2.5倍的最小的二进制长度的文本格式) 而储存PAC实体的二进制格式

`wpa_supplicant`允许的最大的一些围绕解决认证服务器的错误交互操作的问题工作的“EAP workaround”数量, 因为目前认证服务器存在大量的某些问题, 所以这些选项默认激活的。严格的一致性EAP模式可以通过使用选项`eap_workaround=0` 取消工作区来配置

### 活动站点的限制

- 如果一个 在`ap_max_inactivity` 时间内 (秒) 没有发送任何东西, 一个空的数据帧将会发送给它来验证它是够还在搜索范围内。如果该数据帧没有收到应答ASK, 这个站点将会被分离然后失去认证。这个特性当STAs移动出某个范围后, 用于清理旧的实体。
- 在AP仍然还在范围内的情况下可以与站点取得关联; 调查活性是一种验证是否存活的一个不错的方法; 即客户端将不会在连接失败时报告, 因为失去连接的信息数据帧不会在STA发送第一次调查帧后马上发送。
- 默认数值为: 300s (即5分钟)
- `ap_max_inactivity=300`  
AP模式下信标间隔的DTIM周期
- `dtim_period=2`
- 信标间隔 (默认为100TU)
- `beacon_int=100`
- MAC地址规则
- 0 = 使用永久MAC地址
- 1 = 使用为每个ESS连接分配随机的MAC地址
- 2 = 类似于参数1 但是使用唯一的标志序列OUI (有当地管理员设置)
- `mac_addr=0`

`disable_ht`: 是否取消HT (802.11n)

- 0 = 使能HT (802.11n) (如果AP支持的话)

- 1 = 取消  
disable\_ht40 : 是否取消HT-40 (802.11n)
- 0 = 使能HT-40 (802.11n) (如果AP支持的话)
- 1 = 取消HT-40 (802.11n)  
disable\_sgi : 是否取消SGI (short guard interval短护栏间隔)
- 0 = 激活SGI (如果AP支持的话)
- 1 = 取消SGI  
disable\_ldpc : 是否取消LDPC
- 0 = 激活LDPC (如果AP支持的话)
- 1 = 取消LDPC  
ht40\_intolerant : 是否取消容忍40MHZ
- 0 = 容忍40MHZ
- 1 = 不容忍40MHZ  
\* ht\_mcs : 配置为允许MCS率\*
- 解析为一个十六进制的字节数组
- ht\_mcs=" " // Use all available (default)默认使用所有可用的数值
- ht\_mcs=" 0xff 00 00 00 00 00 00 00 00 " // Use MCS 0-7 only只使用0-7MCS
- ht\_mcs=" 0xff ff 00 00 00 00 00 00 00 " // Use MCS 0-15 only只使用0-15MCS  
disable\_max\_amsdu : 是否取消MAX\_AMSDU
- -1 = 不做任何改变
- 0 = 如果硬件支持的话使能MAX\_AMSDU
- 1 = 取消 AMSDU  
ampdu\_factor : 外部最大的A-MPDU长度
- 数值范围: 0-3, see 7.3.2.56.3 in IEEE Std 802.11n-2009.  
\* ampdu\_density : 允许重写AMPDU的密度配置\*
- 作为提示被内核处理
- -1 = 不做任何改变。
- 0-3 = 设置AMPDU的密度 (及称为因子) 为指定的值  
disable\_vht : 是否取消VHT
- 0 = 激活VHT (如果AP支持的话)
- 1 = 取消VHT  
vht\_capa : 设置VHT为重写功能  
vht\_capa\_mask : VHT功能的掩码  
vht\_rx\_mcs\_nss\_1/2/3/4/5/6/7/8: override the MCS set for RX NSS 1-8重写MCS设置为RX NSS 1-8  
vht\_tx\_mcs\_nss\_1/2/3/4/5/6/7/8: override the MCS set for TX NSS 1-8重写MCS设置为TX MSS 1-8
- 0 = MCS 0-7
- 1 = MCS 0-8
- 2 = MCS 0-9
- 3 = 不支持

## 支持快速会话转移

- 本节的选项只在当编译hostapd时设置建立配置选项CONFIG\_FST。这些选项允许这个接口成为FST步骤的一部分。
- FST是从相同的或者不同的一个通道到另外一个通道的会话转移。

- 更多细节可以查看 IEEE Std 802.11ad-2012.

## 接口所属的FST标志组

- fst\_group\_id=bond0

## FST组的接口优先级

- 定义一个接口的更高的优先级意味着FST转换器拥有更高的优先级
- fst\_priority的取值范围为：1-255，当值为1时拥有最低优先级
- fst\_priority=100  
这个接口的默认LLT值（ms）这个值被用于会话建立后没有提供数值的情况，默认为50ms
- fst\_llt 值的范围是1- 4294967（以为spec的限制，查看10.32.2.2 状态之间的转换）
- fst\_llt=100

---

以下为网络语句块的例程：

简单的案例： WPA-PSK, PSK ASCII密文，允许所有有效密码

```
network={
    ssid="simple"
    psk="very secret passphrase"
    priority=5
}
```

和以上类似，增加了请求SSID的特定扫描（用于拒绝广播SSID 的APs）

```
network={
    ssid="second ssid"
    scan_ssid=1
    psk="very secret passphrase"
    priority=2
}
```

只在WPA-PSK时使用，接受任何可用的密码组合

```
network={
    ssid="example"
    proto=WPA
    key_mgmt=WPA-PSK
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk=06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c72ac7bb
    priority=2
}
```

使用TKIP加密算法的WPA-Personal (PSK)，并且强制频繁地重置PTK密码

```
network={
    ssid="example"
    proto=WPA
    key_mgmt=WPA-PSK
    pairwise=TKIP
    group=TKIP
    psk="not so secure passphrase"
    wpa_ptk_rekey=600
}
```

当使用WPA-EAP时。允许CCMP和TKIP加密算法。使用WEP104或者WEP40作为组密码将不会被接受

```
network={
    ssid="example"
    proto=RSN
    key_mgmt=WPA-EAP
    pairwise=CCMP TKIP
    group=CCMP TKIP
    eap=TLS
    identity="user@example.com"
    ca_cert="/etc/cert/ca.pem"
    client_cert="/etc/cert/user.pem"
    private_key="/etc/cert/user.prv"
    private_key_passwd="password"
}
```



```
    priority=1
}
```

使用新的peaplabel1标签来进行RADIUS 服务器的EAP-PEAP/MSCHAPv2 配置（例如，辐射器）

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="user@example.com"
    password="foobar"
    ca_cert="/etc/cert/ca.pem"
    phase1="peaplabel=1"
    phase2="auth=MSCHAPV2"
    priority=10
}
```

使用非加密的匿名标志的EAP-TTLS/EAP-MD5-Challenge配置，真实的标识符只在加密的TLS隧道上发送

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="user@example.com"
    anonymous_identity="anonymous@example.com"
    password="foobar"
    ca_cert="/etc/cert/ca.pem"
    priority=2
}
```

使用非加密的匿名标志的EAP-TTLS/MSCHAPv2配置，真实的标识符只在加密的TLS隧道上发送

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="user@example.com"
    anonymous_identity="anonymous@example.com"
    password="foobar"
    ca_cert="/etc/cert/ca.pem"
    phase2="auth=MSCHAPV2"
}
```

用不同的CA认证证书搭建WPA-EAP，EAP-TTLS 外部和内部认证

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=TTLS
    # Phase1 / outer authentication
    anonymous_identity="anonymous@example.com"
    ca_cert="/etc/cert/ca.pem"
    # Phase 2 / inner authentication
    phase2="auth=TLS"
    ca_cert2="/etc/cert/ca2.pem"
    client_cert2="/etc/cer/user.pem"
    private_key2="/etc/cer/user.prv"
    private_key2_passwd="password"
    priority=2
}
```

支持WPA-PSK 和 WPA-EAP。只有接受CCMP作为成对分组密码

```
network={
    ssid="example"
    bssid=00:11:22:33:44:55
    proto=WPA RSN
    key_mgmt=WPA-PSK WPA-EAP
    pairwise=CCMP
    group=CCMP
    psk=06b4be19da289f475aa46a33cb793029d4ab3db7a23ee92382eb0106c72ac7bb
}
```

SSID 中的特殊字符，使用十六进制。默认为WPA-PSK，WPA-EAP和所有可用的密文

```
network={
    ssid=00010203
    psk=000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
}
```

## GSM SIM 或者 USIM 的EAP-SIM

```
network={
    ssid="eap-sim-test"
    key_mgmt=WPA-EAP
    eap=SIM
    pin="1234"
    pscsc=""
}
```

## EAP-PSK

```
network={
    ssid="eap-psk-test"
    key_mgmt=WPA-EAP
    eap=PSK
    anonymous_identity="eap_psk_user"
    password=06b4be19da289f475aa46a33cb793029
    identity="eap_psk_user@example.com"
}
```

IEEE 802.1x/eapol动态生成WEP密钥（即没有WPA）使用EAP-TLS认证和密钥生成；需要单播和广播WEP密钥。

```
network={
    ssid="1x-test"
    key_mgmt=IEEE8021X
    eap=TLS
    identity="user@example.com"
    ca_cert="/etc/cert/ca.pem"
    client_cert="/etc/cert/user.pem"
    private_key="/etc/cert/user.prv"
    private_key_passwd="password"
    eapol_flags=3
}
```

## LEAP动态WEP密匙

```
network={
    ssid="leap-example"
    key_mgmt=IEEE8021X
    eap=LEAP
    identity="user"
    password="foobar"
}
```

## EAP-IKEv2 使用为服务器和对等认证共享密码

```
network={
    ssid="ikev2-example"
    key_mgmt=WPA-EAP
    eap=IKEV2
    identity="user"
    password="foobar"
}
```

## EAP-FAST with WPA (WPA or WPA2)

```
network={
    ssid="eap-fast-test"
    key_mgmt=WPA-EAP
    eap=FAST
    anonymous_identity="FAST-000102030405"
    identity="username"
    password="password"
    phase1="fast_provisioning=1"
    pac_file="/etc/wpa_supplicant.eap-fast-pac"
}

network={
    ssid="eap-fast-test"
    key_mgmt=WPA-EAP
```

```

eap=FAST
anonymous_identity="FAST-000102030405"
identity="username"
password="password"
phase1="fast_provisioning=1"
pac_file="blob://eap-fast-pac"
}

```

### 纯文本连接(no WPA, no IEEE 802.1X)

```

network={
    ssid="plaintext-test"
    key_mgmt=NONE
}

```

### \* Shared WEP key connection共享WEP密钥连接 (no WPA, no IEEE 802.1X) \*

```

network={
    ssid="static-wep-test"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
}

```

### Shared WEP key connection共享WEP密钥连接(no WPA, no IEEE 802.1X) 使用IEEE 802.11 认证共享密钥

```

network={
    ssid="static-wep-test2"
    key_mgmt=NONE
    wep_key0="abcde"
    wep_key1=0102030405
    wep_key2="1234567890123"
    wep_tx_keyidx=0
    priority=5
    auth_alg=SHARED
}

```

### IBSS/ad-hoc network with RSN

```

network={
    ssid="ibss-rsn"
    key_mgmt=WPA-PSK
    proto=RSN
    psk="12345678"
    mode=1
    frequency=2412
    pairwise=CCMP
    group=CCMP
}

```

### IBSS/ad-hoc network with WPA-None/TKIP (deprecated已经过时了的)

```

network={
    ssid="test adhoc"
    mode=1
    frequency=2412
    proto=WPA
    key_mgmt=WPA-NONE
    pairwise=NONE
    group=TKIP
    psk="secret passphrase"
}

```

### 开放匹配的网络

```

network={
    ssid="test mesh"
    mode=5
    frequency=2437
    key_mgmt=NONE
}

```

### secure (SAE + AMPE) network

```
network={
    ssid="secure mesh"
    mode=5
    frequency=2437
    key_mgmt=SAE
    psk="very secret passphrase"
}
```

## 涵盖所有允许更多或者更少的配置模式的案例

```
network={
    ssid="example"
    scan_ssid=1
    key_mgmt=WPA-EAP WPA-PSK IEEE8021X NONE
    pairwise=CCMP TKIP
    group=CCMP TKIP WEP104 WEP40
    psk="very secret passphrase"
    eap=TTLS PEAP TLS
    identity="user@example.com"
    password="foobar"
    ca_cert="/etc/cert/ca.pem"
    client_cert="/etc/cert/user.pem"
    private_key="/etc/cert/user.prv"
    private_key_passwd="password"
    phase1="peaplabel=0"
}
```

## EAP-TLS智能卡的案例（openssl的工具）

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=TLS
    proto=RSN
    pairwise=CCMP TKIP
    group=CCMP TKIP
    identity="user@example.com"
    ca_cert="/etc/cert/ca.pem"
    client_cert="/etc/cert/user.pem"

    engine=1
    //这里配置的工具必须是可用的。在全局选项中查看openssl支持的工具。
    //通过该引擎的密钥必须和配置上述的客户端证书的私钥匹配

    //使用opensc工具
    //engine_id="opensc"
    //key_id="45"

    //使用pkcs11 工具
    engine_id="pkcs11"
    key_id="id_45"

    //PIN可选的配置，这个选项可以被遗漏并且PIN将会被控制接口请求
    pin="1234"
```

## 如何使用内联blob作为CA认证证书的数据取代使用外部文件来配置的案例

```
network={
    ssid="example"
    key_mgmt=WPA-EAP
    eap=TTLS
    identity="user@example.com"
    anonymous_identity="anonymous@example.com"
    password="foobar"
    ca_cert="blob://exampleblob"
    priority=20
}
blob-base64-exampleblob={
SGVsbG8gV29ybGQhCg==
}
```

## 对于SSID通配符的匹配（只支持纯文本APs）在本例中无论SSID为和都选择任何开放的AP

```
network={
    key_mgmt=NONE
}
```

## 配置两个在这个网络中会被忽略的APs 黑名单

```
network={
    ssid="example"
    psk="very secret passphrase"
    bssid_blacklist=02:11:22:33:44:55 02:22:aa:44:55:66
}
```

该案例配置了AP选择时指定特定APs的限制；任何其他不匹配掩码地址的AP将会被忽略

```
network={
    ssid="example"
    psk="very secret passphrase"
    bssid_whitelist=02:55:ae:bc:00:00/ff:ff:ff:ff:00:00 00:00:77:66:55:44/00:00:ff:ff:ff:ff
}
```

## 只扫描通道36 的配置案例

```
freq_list=5180
network={
    key_mgmt=NONE
}
```

## MACsec配置案例

```
network={
    key_mgmt=IEEE8021X
    eap=TTLS
    phase2="auth=PAP"
    anonymous_identity="anonymous@example.com"
    identity="user@example.com"
    password="secretr"
    ca_cert="/etc/cert/ca.pem"
    eapol_flags=0
    macsec_policy=1
}
```

顶

0

踩

0

---

---

- [上一篇wpa\\_supplicant 介绍（译文）](#)
- [下一篇PID控制算法推算](#)

## 参考知识库

猜你在找

查看评论

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

个人资料



[qq\\_22716879](#)



- 访问: 11620次
- 积分: 417
- 等级: 
- 排名: 千里之外
- 原创: 24篇
- 转载: 14篇
- 译文: 6篇
- 评论: 0条

## 文章搜索

## 文章分类

- [Linux嵌入式开发](#) (12)
- [C/C++](#) (3)
- [QT](#) (8)
- [QML](#) (3)
- [嵌入式课程](#) (5)
- [控制算法](#) (1)
- [git](#) (1)
- [node.js](#) (2)
- [通信/网络](#) (4)
- [WebRTC](#) (3)

## 文章存档

- [2016年12月](#) (6)
- [2016年09月](#) (2)
- [2016年07月](#) (4)
- [2016年06月](#) (2)
- [2016年05月](#) (9)
- [2016年04月](#) (3)
- [2016年03月](#) (16)

## 阅读排行

- [wpa\\_supplicant 的配置说明文件 wpa\\_supplicant.conf 译文](#) (4659)
- [qmake: could not exec '/usr/lib/x86\\_64-linux-gnu/qt4/bin/qmake': No such file or directory](#) (1091)
- [./stdio.h:1010:1: error: 'gets' undeclared here \(not in a function\)](#) (671)
- [Qt通过api获取天气信息](#) (527)
- [PID控制算法推算](#) (456)
- [用户空间GPIO的调用](#) (370)
- [NFS作为根文件系统时server not responding问题的解决](#) (219)
- [/usr/bin/env: node: No such file or directory](#) (203)
- [wpa\\_supplicant 介绍 \(译文\)](#) (196)
- [cp命令出现omitting directory的解决方法](#) (194)

## 评论排行

- [WebRTC Native APIs](#) (0)
- [进程控制开发](#) (0)
- [串口通讯流控制](#) (0)
- [vim中的配置](#) (0)
- [文件I/O编程](#) (0)
- [理解int main\(int argc, char \\*\\*argv\)中的参数的意义](#) (0)
- [原来sscanf函数可以这么强大](#) (0)
- [对话Linus Torvalds: 大多黑客甚至连指针都未理解](#) (0)
- [ioctl\(\)函数详解](#) (0)
- [vim打开多窗口、多文件之间的切换](#) (0)

## 推荐文章

- [\\* 而立之年——三线城市程序员的年终告白](#)
- [\\* Java集合框架中隐藏的设计套路](#)

- o [\\* Python脚本下载今日头条视频\(附加Android版本辅助下载器\)](#)
- o [\\* 人工智能的冷思考](#)
- o [\\* React Native 实战系列教程之热更新原理分析与实现](#)