MDK 3.0 v6 - “是的，好吧，无论如何”通过k2wrlz的ASPj，使用aircrack-ng的osdep库并得到了伟大的aircrack-ng社区的大力帮助：

Antragon，moongray，Ace，Zero\_Chaos，Hirte，thefkboss，ducttape，

telek0miker，Le\_Vert，sorbo，Andy Green，bahathir和Dawid Gajownik

谢谢！

MDK是利用常见IEEE 802.11协议弱点的概念验证工具。

重要：您有责任确保您有权限

运行MDK之前的网络所有者。

该代码是根据GPLv2授权的

MDK用途：

mdk3 <interface> <test\_mode> [test\_options]

尝试mdk3 --fullhelp的所有测试选项

尝试mdk3 --help <test\_mode>仅用于一个测试的信息

**b – 信标泛滥模式**

      发送信标帧以在客户端显示假AP。这有时可能会使网络扫描仪甚至驱动程序崩溃！

 -n <ssid>

         使用SSID <ssid>而不是随机生成的

      -f <filename>

         从文件读取SSID

      -v <filename>

         从文件读取MAC和SSID。查看示例文件！

      -d

         显示站为Ad-Hoc

      -w

         设置WEP位（生成加密网络）

      -G

         显示站为54Mbit

      -t

         显示站使用WPA TKIP加密

      -一个

         显示站使用WPA AES加密

      -m

         使用OUI数据库中的有效接入点MAC

      -H

         跳到AP欺骗的地方 ，这使得测试对某些设备/驱动程序更有效， 但是由于信道跳跃，它降低了分组速率。

      -c <chan>

         在频道<chan>上伪造AP。如果你想要你的卡片，这个频道，你必须设置-h选项，也是！

      -s <pps>

         设置每秒数据包的速度（默认值：50）

**a - 验证DoS模式**

      向认证范围内的所有AP发送认证帧 。 太多客户端冻结或重置某些AP。

-a <ap\_mac>

          测试指定BSSID

       -m

          从OUI数据库使用有效的客户端MAC

       -C

          不要检查测试是否成功

       -i <ap\_mac>

          对AP执行智能测试（-a和-c将被忽略）

          此测试将客户端连接到AP，并重新注入嗅探数据以保持其活动

       -s <pps>

          设置每秒包的速度（默认：无限制）

**p - 基本探测AP信息和ESSID猜解模式**

      探测器AP并检查答案，用于检查SSID是否有用

      如果AP处于适配器的发送范围内，

      SSID也可以通过此测试模式强制实施。

-e <ssid>

          告诉mdk3哪个SSID要探测

       -f <filename>

          从文件读取行以强制隐藏的SSID

       -t <bssid>

          设置目标AP的MAC地址

       -s <pps>

          设置速度（默认值：无限制，以Bruteforce模式：300）

       -b <字符集>

          使用完整的Bruteforce模式（仅推荐用于短SSID！）

          仅使用此开关显示其帮助屏幕。

**d - 去认证/解除关联Amok模式**

-w <filename>

          读取包含MAC的文件不关心（白名单模式）

       -b <filename>

          读取包含MAC的文件以运行测试（黑名单模式）

       -s <pps>

          设置每秒包的速度（默认：无限制）

       -c [chan，chan，chan ...]

          启用频道跳频 没有提供任何渠道，mdk3将跳过一切

          14个b / g通道。 频道将每5秒更改一次。

      踢所有人从AP找到

**m - 迈克尔关闭利用（TKIP）**

       全部取消所有流量

       -t <bssid>

          设置目标AP的Mac地址

       -w <秒>

          突发之间的秒数（默认值：10）

       -n <ppb>

          每个突发设置数据包（默认值：70）

       -j

          使用新的TKIP QoS-Exploit

          只需要几个数据包就可以关闭AP了！

       -s <pps>

          设定速度（默认值：400）      全部取消所有流量

**x - 802.1X测试**

-n <ssid>

使用SSID <ssid>

-t <bssid>

设置目标AP的MAC地址

 -w <WPA类型>

设置WPA类型（1：WPA，2：WPA2 / RSN;默认值：WPA

-u <unicast cipher>

设置单播密码类型（1：TKIP，2：CCMP;默认值：TKIP）

-m <组播密码>

设置组播密码类型（1：TKIP，2：CCMP;默认值：TKIP）

-s <pps>

设定速度（默认值：400

1 - EAPOL注销测试

-t <bssid>

设置目标AP的MAC地址

-c <bssid>

设置目标STA的MAC地址

-s <pps>

设定速度（默认值：400）

**w - WIDS / WIPS / WDS混乱**

       将WDS与多个身份验证的客户端混淆，从而混淆路由表

       -e <SSID>

          目标WDS网络的SSID

       -c [chan，chan，chan ...]

          使用频道跳频

       -z

          激活Zero\_Chaos的WIDS漏洞

          （认证客户从WDS到外国APs，让WIDS坚果）

**f - MAC过滤器bruteforce模式**

       此测试使用已知客户端MAC地址的列表并尝试

       在给定的AP进行动态更改时将其认证

       其响应超时以获得最佳性能。 它目前只能工作

       拒绝正确打开认证请求的AP

       -t <bssid>

          目标BSSID

       -m <mac>

          将MAC地址范围设置为使用（3字节，即00:12:34）

          没有-m，将使用内部数据库

       -f <mac>

          设置MAC地址以开始强制

          （注意：您不能同时使用-f和-m）

**g - WPA降级测试**

       解密站点和AP发送WPA加密数据包。

       通过这个测试你可以检查系统管理员是否会尝试设置他的

       网络到WEP或禁用加密。 mdk3会让WEP和加密

       客户端工作，所以如果系统管理员简单地认为“WPA坏了”他

       肯定不是这个工作的正确的。

       （这可以/应该与社会工程相结合）

       -t <bssid>

          目标网络