Misc-杂项破解

2021-09-28

当遇到有密码的地方的时候，直接暴力破解一把梭往往是最优解呢XD

# Misc-杂项破解

## WIFI密码破解

### [Aircrack-ng](http://www.aircrack-ng.org/)

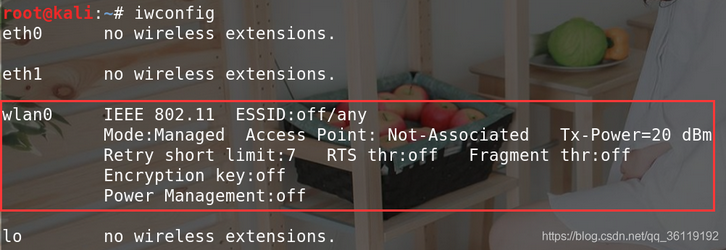
Aircrack- ng 是一套完整的评估 WiFi 网络安全的工具。它专注于 WiFi 安全的不同领域：

* 监控：数据包捕获和数据导出到文本文件以供第三方工具进一步处理
* 攻击：通过数据包注入重放攻击、解除身份验证、虚假接入点等
* 测试：检查 WiFi 卡和驱动程序功能（捕获和注入）
* 破解：WEP 和 WPA PSK（WPA 1 和 2）

所有工具都是命令行，允许编写大量脚本。 许多 GUI 都利用了此功能。它主要适用于 Linux，但也适用于 Windows、macOS、FreeBSD、OpenBSD、NetBSD，以及 Solaris 甚至 eComStation 2。

Aircrack-ng已经集成在Kali之中，可以直接使用。故以下流程基于Kali平台进行演示。

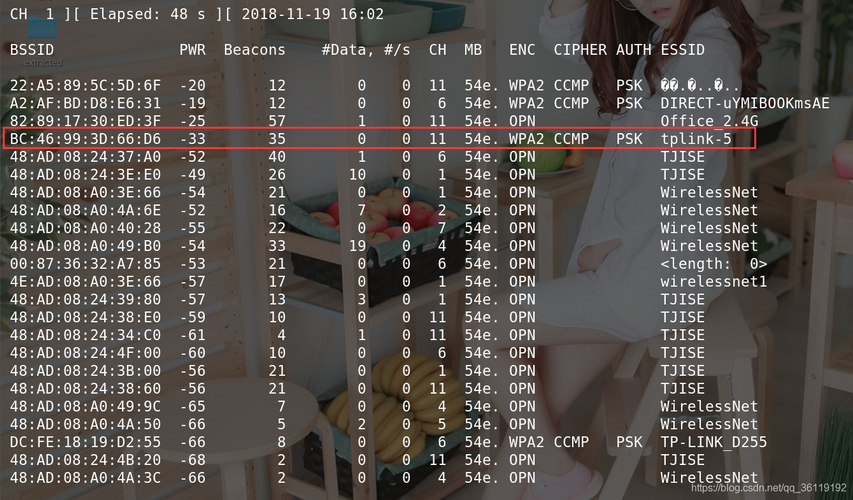
1. 运行iwconfig命令检查网络信息。在使用无线网卡的情况下，应该能看到以下信息：

* 

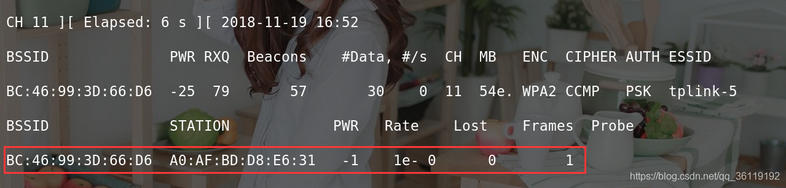
1. 使用airmon-ng工具，启动无线网卡的监听模式（前提是无线网卡支持此功能）

* airmon-ng start wlan0
* 
* 注意，网卡开启了监听模式之后网卡的名字就变为 wlan0mon 了，以后的所有命令中的网卡名都是叫 wlan0mon

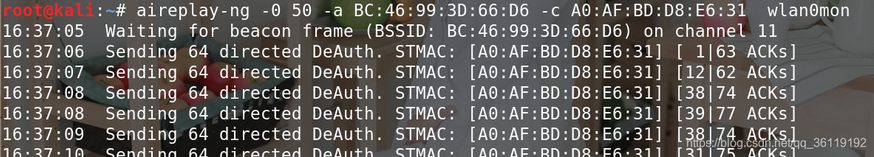
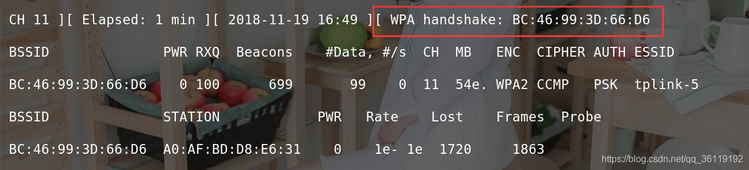
1. 扫描附近的WIFI

* airodump-ng wlan0mon
  + BSSID代表路由器的 MAC 地址
  + PWR 代表信号的强度，数值越大代表信息越强
  + CH 代表信道
  + ENC代表用的加密的方式
  + AUTH 代表认证的方式
  + ESSID是WIFI的名字
* 

1. 确定一个要攻击的网络，然后开始监听流量：

* airodump-ng -w tplink -c 11 --bssid BC:46:99:3D:66:D6 wlan0mon
  + -w 参数指定生成的文件名
  + -c 指定信道
  + --bssid指定路由器的MAC地址
* 
* 注意，在输出中，可以看到目前的连接设备，在抓包过程中，应当选择活跃的设备（上图的STATION），以便于后面数据包的获取。

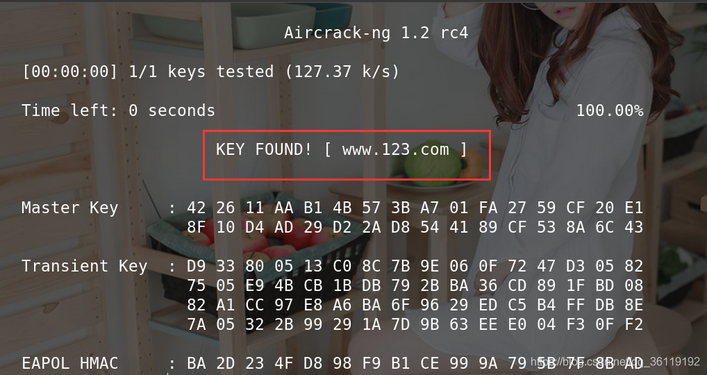
1. 开始攻击客户端与路由器的连接，以抓取连接的握手包：

* 另外开一个terminal，开始攻击连接，同时保持原来的监听窗口，方便监控连接数据：
* aireplay-ng -0 50 -a BC:46:99:3D:66:D6 -c A0:AF:BD:D8:E6:31 wlan0mon
  + 50是发包的数量
  + -a指定路由器的MAC地址
  + -c指定连接的客户端的MAC地址
* 
* 该命令会打断客户端和路由器之间的连接，等到客户端重新连接WIFI的时候，抓包进程就能获取到握手包的数据。
* 
* 如上图，如发现了出现WPA handshake的输出，则说明已经成功抓取了客户端和路由器之间的握手数据，其中就保存着WIFI的连接密码。

1. 在上面的步骤中，由于定义了输出位置，会在本地生成四个文件，其中包含.cap流量数据：

* 
* 接下来，就是对抓取到的cap包进行暴力破解，这需要准备好一份用于爆破的密码字典。一个强大的字典，就意味着爆破出密码的可能性被大大提高。Kali下自带有一份无线密码字典——> /usr/share/wordlists/rockyou.txt.gz ，我们将其解压，其中rockyou.txt就是密码文件。同时，也可以使用[Github-PasswordDic](https://github.com/k8gege/PasswordDic)此开源项目整理的密码字典进行爆破。
* aircrack-ng -w /usr/share/wordlists/rockyou.txt -b BC:46:99:3D:66:D6 tplink-01.cap
  + -w 指定密码字典
  + -b指定路由器的MAC地址

1. 如果字典文件中存在WIFI的密码的话，只需要耐心等待，就可以得到正确密码：

* 
* WIFI密码也就是：www.123.com

## Office密码破解

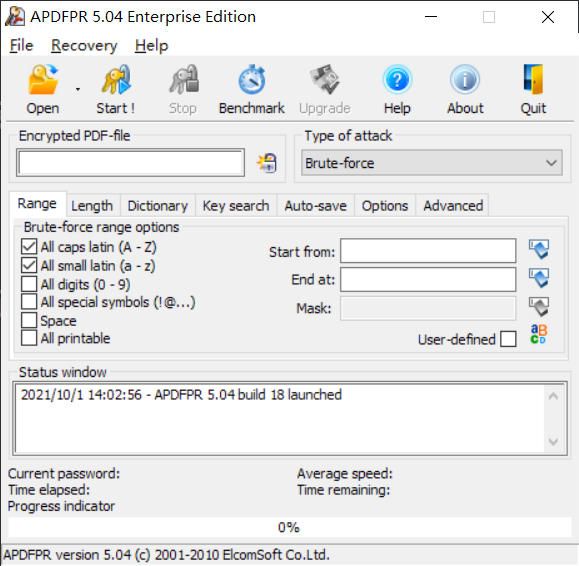
### Office Password Remover

傻瓜式软件，按照向导指定文件即可移除文件的密码保护。

## PDF密码破解

### Advanced PDF Password Recovery

Advanced PDF Password Recovery同ARCHPR一样，出自[elcomsoft](https://cn.elcomsoft.com)，甚至界面都如出一辙。



选择好参数，选择Start！就好了。操作流程与ARCHPR大同小异。

## 其余密码破解

### john

John the Ripper 是一个快速的密码破解器，目前可用于 Unix、macOS、Windows、DOS、BeOS 和 OpenVMS（后者 需要一个贡献的补丁）。 其主要目的是检测弱 Unix 密码。 除了几种 crypt(3) 密码哈希类型之外，大多数 常见于各种 Unix 版本，开箱即用的支持是 Kerberos/AFS 和 Windows LM 散列，以及基于 DES 的行程码，以及 “-jumbo”版本中的数百个附加哈希和密码。

### hydra

Number one of the biggest security holes are passwords, as every password security study shows.  
This tool is a proof of concept code, to give researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.

hydra主要是针对服务的攻击，对于像https这类的加密连接，和简单的网页登陆的弱口令爆破，建议使用Burp Suite

## 哈希破解

Hash，一般翻译做散列、杂凑，或音译为哈希，是把任意长度的输入（又叫做预映射pre-image）通过散列算法变换成固定长度的输出，该输出就是散列值。这种转换是一种压缩映射，也就是，散列值的空间通常远小于输入的空间，不同的输入可能会散列成相同的输出，所以不可能从散列值来确定唯一的输入值。简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的函数。

常见的哈希函数有MD4（MD5的前身），MD5（比赛中最为常见），以及SHA家族，并且SHA家族的抗穷举性比MD5更好。这里做个比较：

original text: Hello World  
 MD5: b10a8db164e0754105b7a99be72e3fe5  
 SHA-1: 0a4d55a8d778e5022fab701977c5d840bbc486d0  
 SHA\_256: a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e

由于加密后的密文长度更长，导致哈希碰撞的复杂度在SHA家族上呈极大增加，彩虹表的思路在SHA家族上已经不能发挥如同在MD5上面的效果。故这里哈希破解只针对于MD5的破解。

### MD5爆破

彩虹表查询：<https://www.cmd5.com>

密码学计算工具：<https://gchq.github.io/CyberChef/>

由于其他语言实现MD5计算存在难度，故以下脚本皆使用Python攥写。

#### MD5的Python计算

在Python中，已存在包含MD5计算函数的库：hashlib，以下为代码示例：

import hashlib  
m = hashlib.md5()  
m.update(b'123')  
s=m.hexdigest()  
print(s)  
>>> "202cb962ac59075b964b07152d234b70"

#### 确定长度类

确定了长度也就确定了暴力破解的可能性。这里建议使用Python中的string库生成字符字典：

import string  
dic\_number=string.digits  
>> "0123456789"  
dic\_letter=string.ascii\_letters  
>> "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"  
dic\_common=string.digits+string.ascii\_letters+"+-\*/\_"  
>> "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ+-\*/\_"

##### 已知MD5值及salt值

例如，已经知道明文长度为4，并且已知加盐的值和加盐后的MD5，那么就可以爆破。

Exp:
  
MD5(salt+text):0b9d78b208dbeaca4dec33e5ebe9a863
  
salt:12345
  
text=flag
  
len(text):4

已知以上信息，便可以编写脚本破解：

import string  
import hashlib  
dic\_common = string.digits+string.ascii\_letters+"+-\*/\_"  
md5 = "0b9d78b208dbeaca4dec33e5ebe9a863"  
salt = "12345"  
for letter\_1 in dic\_common:  
 for letter\_2 in dic\_common:  
 for letter\_3 in dic\_common:  
 for letter\_4 in dic\_common:  
 pw = salt+letter\_1+letter\_2+letter\_3+letter\_4  
 s=hashlib.md5(pw.encode()).hexdigest()  
 if s==md5:  
 print(letter\_1+letter\_2+letter\_3+letter\_4)

输出了“flag”，即为明文。

##### 已知字符串构成及MD5值的开头

Exp：
  
md5: 1f3dd1cad1db50769c07cfa62b89c08b
  
text:flag{md5\_is\_fun}
  
  
question:text=flag{md?\_i?\_f??} and md5(text)[0,7]=1f3dd1c

编写脚本求解：

import string  
import hashlib  
dic\_common = string.digits+string.ascii\_letters+"+-\*/\_"  
knownMd5 = "1f3dd1c"  
for letter\_1 in dic\_common:  
 for letter\_2 in dic\_common:  
 for letter\_3 in dic\_common:  
 for letter\_4 in dic\_common:  
 pw = "flag{md%s\_i%s\_f%s%s}"%(letter\_1,letter\_2,letter\_3,letter\_4)  
 s=hashlib.md5(pw.encode()).hexdigest()  
 if s[0:7]==knownMd5:  
 print(pw)

输出：

flag{md5\_is\_fun}

#### 截断验证码

这里直接使用De1CTF线上赛-Web3来举例，打开靶场，看到以下提示：

Code(substr(md5(?),0,5)===9331c)

代码逻辑是将输入的数据进行MD5计算，然后比对前五位是否为"9331c"。那么理论上存在的可能性有16^5=1048576种，页面每次刷新之后md5都会更新一次，但是在现实中，md5前五位相同是多解的。故使用纯数字进行碰撞：

import hashlib  
knownMd5 = '9331c' # 已知的md5明文  
list = []  
def findCode(code):  
 key = code.split(':')  
 start = int(key[0])  
 end = int(key[1])  
 for code in range(start, end):  
 # 计算MD5的值并且比较  
 if hashlib.md5(str(code).encode('utf-8')).hexdigest()[0:5] == knownMd5:  
 print(code)  
 break  
for i in [0,1,2]: # 这里控制爆破出多少结果停止  
 list.append(str(10000000\*i) + ':' + str(10000000\*(i+1))) # 这里开三个区间隔开数据,因为List对象存在大小限制  
for data in list:  
 findCode(data)

输出了三个可能解：

1916798 --> 9331c680d836f30104bbeafc7be471d3  
12032937 --> 9331cd107b8658063844d01cefd5edf4  
21175023 --> 9331ce2a630c3b231e201b18c7ed5509

这三个都可以通过验证。

## 词频爆破

### [quipqiup](http://quipqiup.com/)

quipqiup是Edwin Olson的快速自动密码求解器。它可以解决报纸上经常出现的简单替代密码，包括诸如密码窃听器（保留单词边界）之类的难题和爱国主义者（如密码迷）之类的难题。

quipqiup网站一般用来解决简单替代密码，也就意味着可以解决如凯撒密码，变异凯撒，rot家族这类的密码。因为这类加密方式本质上就是字母一一对应替换，存在无限可能解，但是再运用字母出现频率和语法规则进行筛选，一般就可以求出最优解，往往也就是正确答案。这类解密问题一般给的密文都很长，因为在统计学下，样本越多，对于密文的词频分析会越准确，当然，明文一般也都是有意义的句子。

#### 例题

密文：

MTHJ{CUBCGXGUGXWREXIPOYAOEYFIGXWRXCHTKHFCOHCFDUCGTXZOHIXOEOWMEHZO}

用凯撒密码、rot家族和一些凯撒变种分析，都没有可能解。于是就开始分析可能是词频爆破，用[quipqiup](http://quipqiup.com/)进行分析：



第一个解就是明文，就这样解开了替换密码。

注意，quipqiup工作时，必须提供一定的替换信息，比如MTHJ=FLAG，否则可能无法正确分析。