Um pouco sobre a segurança Wi-Fi - Um Adendo

1 - Comandos utilizados na descoberta de uma rede Wi-Fi WEP:

- ★ Lembrando que eu estava utilizando um sistema GNU/Linux, no caso a distro Slackware 14.2. Se estiver utilizando alguma outra distro poderá ter (mas provavelmente não terá) alguma pequena mudança no nome da interface Wi-Fi (principalmente se utilizar algum dispositivo de WI-FI USB – que são ótimos para se utilizar em uma máquina virtual, e. g., Virtualbox), que neste caso é Wlan0. Os comandos listado aqui tem com base sistemas GNU/Linux. **#** quer dizer comandos devem ser executados em modo root.

- ★ Precisará do programa/*suite* aircrack-ng, que poderá conseguir pelo site do projeto ou em repositórios da distro que estiver utilizando.

- ★ Poderá optar por utilizar a distro Kali Linux (antigo Backtrack), que tem o objetivo de ser uma distro voltada para segurança e *pentest*, que já vem com o aircrack-ng e vários outros programas.

- ★ Poderá instalar o aircrack-ng em um sistema Windows, contudo no Windows não conseguirá habilitar o modo monitor como no GNU/Linux, parte pela WinPcap não dar suporte ao modo monitor, logo terá que comprar algum dispositivo da AirPcap (link_1, link_2). Talvez exista algum outro modo de conseguir o modo monitor no Windows, contudo não me aprofundei muito nesta parte. Assim para esse processo é mais fácil utilizar alguma distro GNU/Linux.

- ★ Se estiver em um Mac OS poderá ter ao modo monitor, contudo com uma pequena diferença no nome da interface sem fio.

Primeiramente é preciso descobrir o nome da minha interface de rede sem fio, posso usar ifconfig, iwconfig ou o airmon-ng (parte do aircrack-ng).
 # airmon-ng

O airmon-ng me diz o nome da interface, neste caso é wlan0.

Agora preciso iniciar esta interface em modo monitor para conseguir capturar “todos” os pacotes que chegam nela.
 # airmon-ng start wlan0

Ao iniciar a interface em modo monitor o aircrack-ng já me avisa de processos que podem me atrapalhar na captura de pacotes, neste caso sendo eles, o NetworkManager, meu gerenciador de conexões e o wpa_supplicant, meu cliente assistente de conexão WPA/2.

Assim preciso matar esses processos. Para tal tenho 3 modos, kill PID (ID do processo), killall nome_do_processo, ou mandar o airmon-ng verificar os processo que podem atrapalhar a captura e caso encontre algum, o mate, comando airmon-ng check kill. Talvez o aircrack-ng também lhe avise sobre o dhcp, mas não precisa se preocupar com ele, ele é executado temporariamente, logo mais será terminado. Se tentar matar o processo dele, provavelmente já estará morto.

 # airmon-ng check kill
ou
 # kill "PID"
ou
 # killall NetworkManager
 # killall wpa_supplicant

Poderá ver a nome da interface "criada" em modo monitor com o ifconfig, iwconfig ou airmon-ng. Com a interface em modo monitor, o airmong-ng criará uma nova interface (virtual) onde posso de fato trabalhar em modo monitor e dependendo da versão irá apenas alterar o nome da interface de rede sem fio, o ocorreu neste caso, wlan0 agora é wlan0mon em modo monitor.

Agora preciso obter informações da rede alvo. O airodump-ng <interface> irá me mostrar várias informações das redes em a minha volta, onde posso ver a rede alvo.
 # airodump-ng wlan0mon

Nesse caso a rede alvo será aquela rede WEP configurado no roteador. Precisamos do BSSID dessa rede, que é 64:66:B3:70:9F:C8, o canal que ele está trabalhando, que é 6. Assim podemos iniciar a captura de pacote desta rede/AP.
 # airodump-ng -c 6 -w wep_test_cap --bssid 64:66:B3:70:9F:C8 wlan0mon

Assim utilizo o airodump-ng para capturar os pacotes, -c para definir o canal de comunicação da rede alvo, no caso 6, -w para salvar tais pacotes em um arquivo com o nome que eu colocar logo a frente, no caso wep_test_cap, depois o bssid da rede, famoso MAC address e, por fim, qual interface irá capturar os pacotes. O airodump-ng irá me mostrar a quantidade de pacotes data já recebidos, em média preciso de 20 a 50 mil pacotes para conseguir quebrar WEP, o que com muito tráfego será poucos minutos.

O airodump-ng criará outros 4 arquivos, mas o importante, no momento, é o com a extensão cap. Baixe o wep_test_cap.cap e também poderá quebrar o WEP.

 # aircrack-ng weptest-01.cap

O aircrack-ng olhará os pacotes capturados com seu IV, ICV e tudo mais e vai tentar quebrar a codificação. Terá como resultado algo similar ao texto abaixo.

```
                                Aircrack-ng 1.2 rc4

                          [00:00:00] Tested 24949 keys (got 72721 IVs)

KB    depth   byte(vote)
 0    0/  1   71(105216) B7(87296) 82(82688) 07(81408) 4C(80384) 60(80384) 80(80384) 92(80384) 05(79872)
 1    0/  1   77(92928) 08(84224) 5A(83712) D7(81920) 92(81664) 54(81408) 05(80896) 72(80896) BA(80896)
 2    0/  1   65(96768) 90(83456) AA(82944) F6(82432) 32(82176) A4(81920) F9(81664) 0A(81408) C5(81408)
 3    0/  1   72(86784) 57(84992) FE(83968) F9(83712) 99(83456) 65(81920) 77(81664) 7D(81664) B4(81664)
 4    0/  1   74(92160) E9(87040) 3C(83712) 74(83456) BC(82944) 4D(82432) E5(82432) 9C(81408) 0C(81152)
 5    0/  1   79(103424) 94(87040) D7(86784) 57(83968) 15(83456) 91(82176) D2(82176) 6E(81408) 12(81152)
 6    0/  1   75(97024) C6(88064) 81(87808) AA(83456) 22(82688) CB(82432) D5(81920) D0(81664) 9D(81408)
 7    0/  1   69(94976) 55(83200) DF(83200) 21(82176) 6F(82176) D0(82176) F0(82176) 80(81920) 97(80896)
 8    0/  1   6F(101120) 0F(85504) 39(84992) 9A(84736) AA(83456) 05(82432) 08(82176) 92(81664) 2D(81408)
 9    0/  1   70(89600) 02(84480) BF(82688) A3(81920) DB(81152) 30(80640) E6(80640) C0(80384) 25(80128)
10    0/  1   84(84480) 2F(82944) 63(82944) B1(82688) 41(81408) EF(81408) 82(80896) F5(80384) 1C(80128)
11    0/  1   B3(86272) 34(83968) 04(83200) 93(82688) F0(82688) 25(80640) 38(80640) E5(80128) 4A(79872)
12    0/  1   64(88060) EC(83484) D6(81536) 65(81172) AC(81148) 4A(80624) 79(80068) 12(79944) 4C(79780)

  KEY FOUND! [ 71:77:65:72:74:79:75:69:6F:70:61:73:64 ] (ASCII: qwertyuiopasd )
     Decrypted correctly: 100%
```

Aircrack-ng 1.2 rc4
[00:00:00] Tested 24949 keys (got 72721 IVs)

KB depth byte(vote)
0 0/ 1 71(105216) B7(87296) 82(82688) 07(81408) 4C(80384) 60(80384)….
……..
KEY FOUND! [71:77:65:72:74:79:75:69:6F:70:61:73:64] (ASCII: qwertyuiopasd)
Decrypted correctly: 100%

E já temos a chave quebrada. Poderá ver o mesmo processo nesse tutorial do Curinga (voz, kkk)

Voltando as interfaces, terá a wlan0mon, eth0 (nome comum para interface ethernet) e a lo (interface loopback), vistas pelo ifconfig.
Nas versões antigas era necessário deletar a interface virtual, que era mon0 (iw dev mon0 del), mas atualmente é o airmon-ng volta a interface para o modo *managed*, facilitando o trabalho. Preste atenção que aqui agora é wlan0mon, nome da interface, que mudou depois do airmon-ng start <interface>.
 # airmon-ng stop wlan0mon

Neste momento eu não tenho nenhum gerenciador de rede ativo, pois matei ele em comandos anteriores, para ativar eles novamente pode variar de distro e de gerenciador de rede, no meu caso é o NetworkManager, que apenas preciso digitar o nome dele para inicializá-lo novamente. Em alguns caso também é preciso iniciar wpa_supplicant, se for conectar em uma rede WPA/2

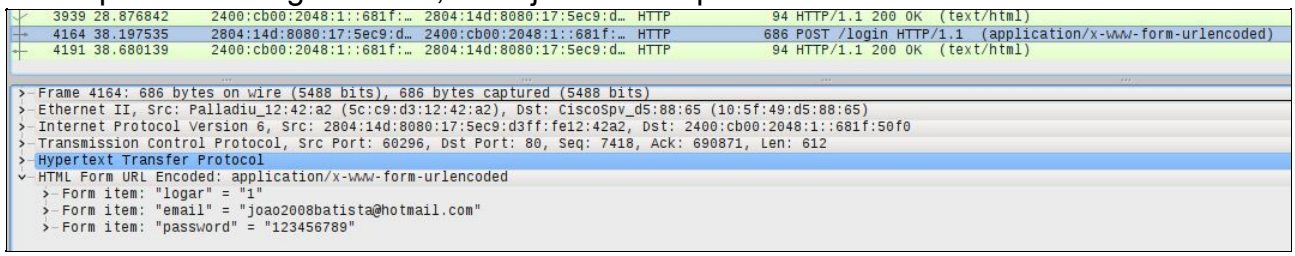
 # NetworkManager
 # wpa_supplicant

Fim da parte da quebra de WEP.

2 - Perguntas interessantes:

1 **Em uma rede com criptografia (WEP, WPA/2), os dados são abertos para alguém que está conectado a rede e está "ouvindo" os pacotes?**
R: A resposta é não. Isso torna tais pacotes seguros? Mais ou menos. Vamos por partes...

Em um teste capturei a minha própria conexão em um site http, unionmangas, e tentei me logar. Na parte de login, este site utiliza um *form* que é enviado via *post* HTTP, tipo de informação que vou reparar nos pacotes capturados. Como era de se esperar, os meus próprios pacotes são "abertos" para mim, ou seja, posso ler o conteúdo deles. Poderá baixar ele, http_post_4164.pcap. Utilize o wireshark para visualizar o pacote. Para melhor visualização apliquem o filtro http e no pacote 4164 verá as informações de login que utilizei para tentar logar no site, ou seja em texto puro.



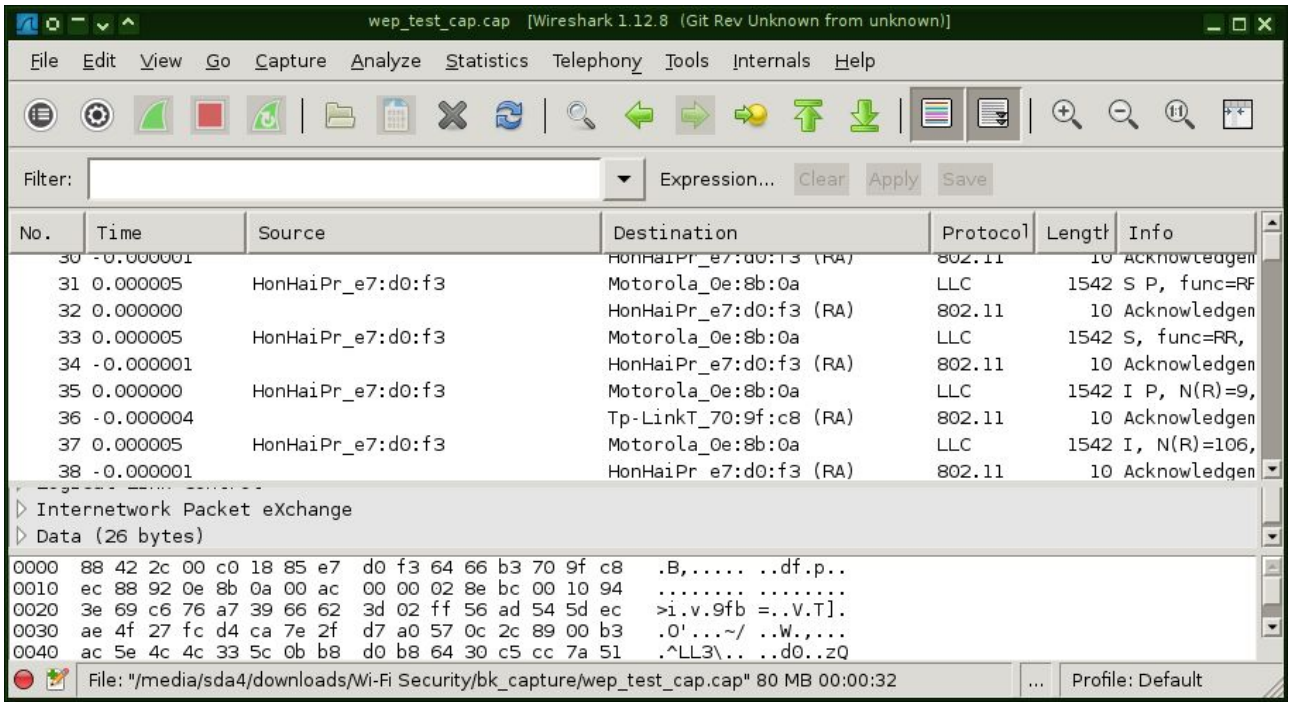
Já o com HTTPS, baixe o https_facebook.pcap, aplique o filtro “ipv6.dst==2a03:2880:f000:1:face:b00c:0:1” (sem as aspas), para limitar a pacotes IPv6 vindos do Facebook, para ver os pacotes trocados com algum servidor do Facebook e meu notebook. Neste caso os dados estão cifrados e precisaria da chave de sessão que o navegador e o Facebook “combinaram” para ver os dados puros. Chave que não é nada fácil de conseguir se não tiver acesso ao meu notebook.

checked

Agora, e se eu capturar pacotes que forem de outras máquinas em uma rede com segurança? Estes pacotes estarão criptografados e se eu estiver utilizando apenas o modo promíscuo, eles serão descartados (jogados no chão, como alguns gostam de dizer), isso ocorre porque os pacotes estão cifrados e a interface de rede então não consegue ler eles. Assim preciso do modo monitor para pegar tais pacotes. Contudo apenas pegar tais pacotes não é o suficiente, pois não vou conseguir ler o conteúdo deles. Para tal preciso de um modo de descriptografar tais pacotes. O Wireshark já tem esta opção, consegue descriptografar eles se estiver habilitado a desencriptação em suas configurações e com as chaves necessárias/certas. Melhor explicar com exemplos.

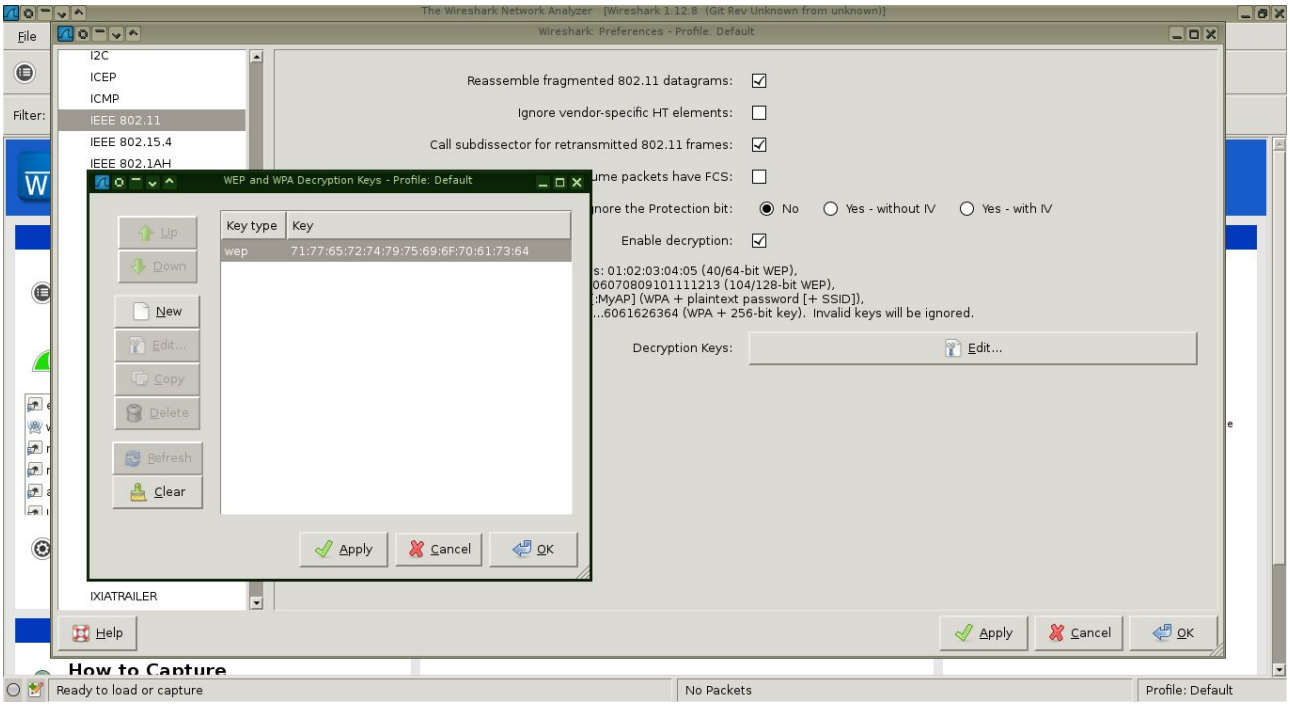
Além das redes cabeadas e sem fio, o wireshark também dá suporte a captura de outros dispositivos/interfaces, como USB e Bluetooth, isso dependendo do seu sistema operacional (veja mais em Network media specific capturing).

Com o pacote WEP capturado anteriormente, abra ele no Wireshark, verá algo como na imagem abaixo, vários pacotes “cifrados”.

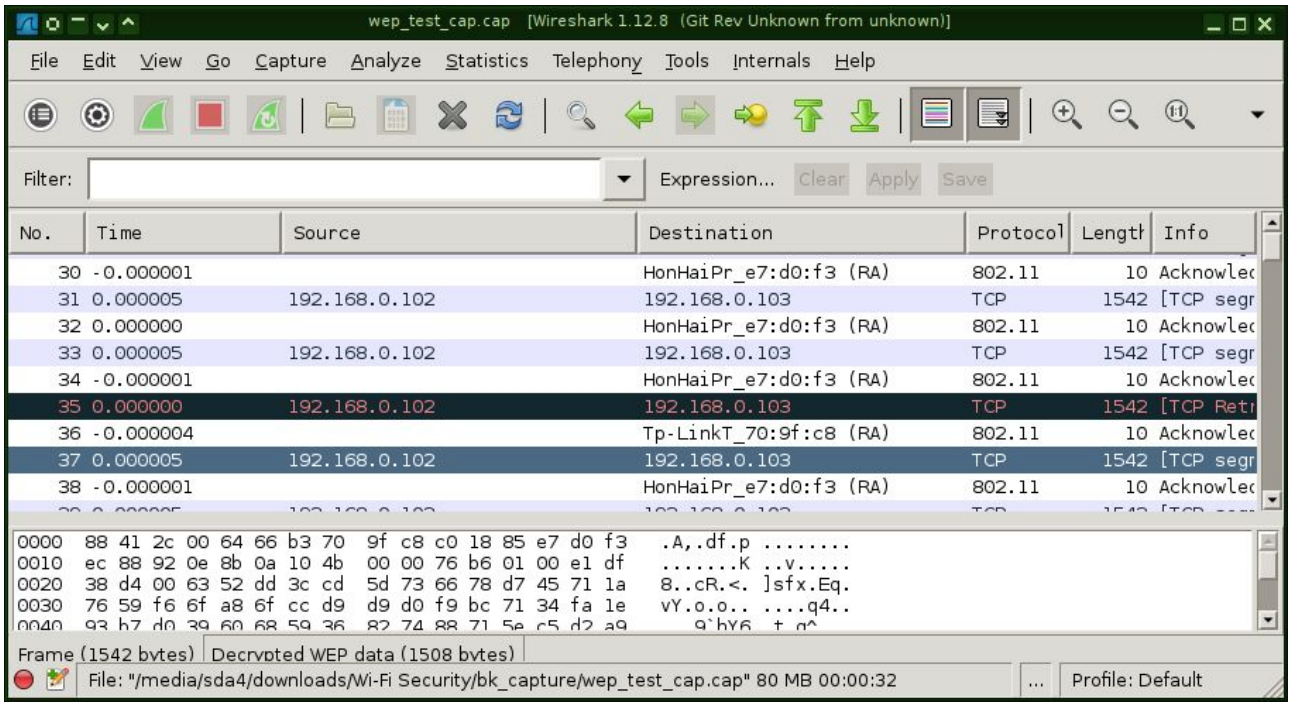


Para descriptografar os pacotes desta captura de rede WEP vou precisar da chave/senha e adicionar ela no Wireshark. A chave já temos, no caso preciso da parte hexadecimal do: KEY FOUND! [71:77:65:72:74:79:75:69:6F:70:61:73:64] (ASCII: qwertyuiopasd), assim

preciso do 71:77:65:72:74:79:75:69:6F:70:61:73:64 e adicionar no wireshark (veja mais sobre em How to Decrypt 802.11) em *Edit->Preferences->IEEE 802.11 ou Edit->Preferences-> Protocol->IEEE 802.11.* Primeiro habilitar decryption e depois adicionar a chave. Como na imagem abaixo:



Agora poderá ver os pacotes descriptografados, como na imagem abaixo:



Se for uma rede WPA-PSK ou WPA2-PSK também vou conseguir descriptografar os pacotes. Se for WPA/2-Entreprise é bem mais complicado.

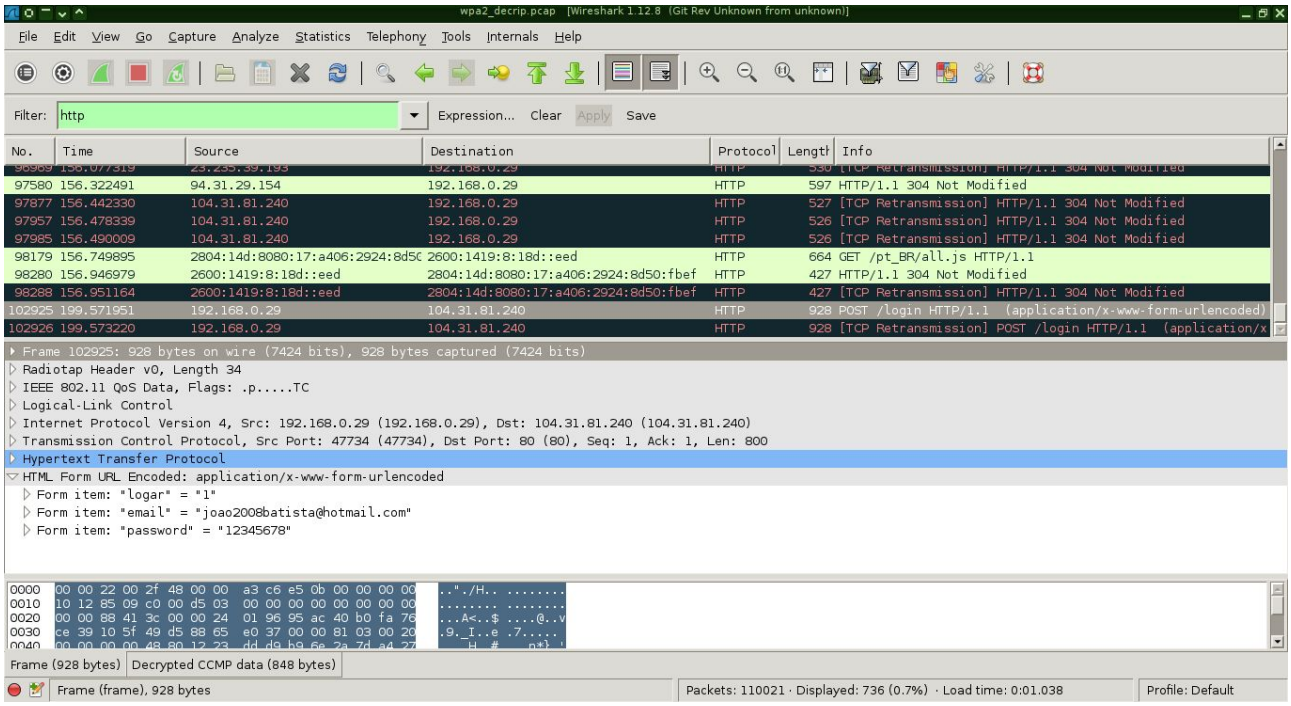
Diferente do WEP, no WPA/2-PSK vou precisar capturar os4-way handshake (aquele esquema de EAP over Lan), sendo esses pacotes: *Auth Request*, *Auth Response*, *Association Request*, *Association Response*. E também ter o nome da rede e sua senha.

Neste link tem um bom exemplo passo a passo.

Agora vamos juntar as duas coisas, captura de http e descriptografia, baixe o wpa2_decrip.pcap.

Nessa captura utilizei uma rede WPA2-PSK da república onde moro e com meu celular tentei logar naquele mesmo site de mangás. Para descriptografar tais os pacotes preciso do nome dessa rede e sua senha, em outras palavras lhe dar acesso a essa rede, contudo também funciona com a WPA-PSK, devolvendo um pouco de segurança para essa tal rede.
A tal chave, gerada pelo PBKDF2 é:
d494a04654c0922aaed4ea24facc9903b03820149d79e4fe9ce7e8fb35fea3f0

Primeiro olhem o arquivo de capturas sem adicionar a chave WPA-PSK e depois com a chave, desta vez o pacote de requisição de login é o 102925. Repare na parte inferior [Frame 928 bytes | Decrypted CCMP data (848 bytes)]



Legal, mas e pra quebrar WPA/2?

Para o WPA vai ser o mesmo para WPA2, vai precisar dos 4-*way handshake* e depois colocar muito processamento em cima deles e da PSK. Existem vários dicionários de palavras, ou de senhas, que podem facilitar esse processo. Um bom tutorial passo a passo.

Arquivos de capturas, o sem os 4-4-way handshake psk1.cap, com os 4 psk2.cap. O wordlist_BIG-WPA-LIST-1 modesto com uma boa quantidade de senhas/palavras e o wordlist com algumas senhas e a "correta" no final. Com o BIG-WPA-LIST-1, irá percorrer todo arquivo de senhas e não vai encontrar a senha correta, porque ela não está no arquivo, mas com o BIG-WPA-LIST-1_s vai ter sucesso.

 # aircrack-ng -w BIG-WPA-LIST-1_s psk-01.cap

Falha por psk-01.cap por não tem os 4-way handshake

```
Opening psk-01.cap
Read 392155 packets.

   #  BSSID              ESSID                     Encryption

   1  64:66:B3:70:9F:C8  topcom2016                WPA (0 handshake)

Choosing first network as target.

Opening psk-01.cap
No valid WPA handshakes found..


Quitting aircrack-ng...
```

 # aircrack-ng -w BIG-WPA-LIST-1_s psk-02.cap
Sucesso e senha encontrada!

```
                          Aircrack-ng 1.2 rc4

 [00:00:04] 9996/9999 keys tested (2135.28 k/s)

 Time left: 0 seconds                                      99.97%

                        KEY FOUND! [ 04102016 ]


 Master Key     : EF 34 7A 43 51 A0 4D AD 0B FC B6 59 DD 99 2C 4C
                  0B 0A 6F D9 97 D6 37 AF B2 72 00 58 2C 52 A7 DB

 Transient Key  : 1F E7 88 DE F6 39 79 7A 9F 81 44 A1 39 B3 E6 BA
                  79 88 7C F0 8D E8 F5 1A E7 E6 A5 C3 53 B5 4B 0C
                  2B 24 11 40 FC 63 7B D1 5F 71 D3 B5 E3 22 5D 7B
                  81 A2 28 0A 07 96 C8 6D 22 16 9A 5B F4 CC 08 57

 EAPOL HMAC     : 95 0C 1A 4E C3 CC E0 27 A6 26 4D 25 12 CC 23 D6
```

E se tiver WPS habilitado?

Com WPS a vida de quem quer descobrir a senha de uma rede WPA/2 fica bem mais fácil, terá que apenas testar por força bruta o PIN usado no WPS.
O Reaver, lhe faz este trabalho. Um bom tutorial passo a passo. Também existem dicionários dos PIN mais comumente utilizados.

- ★ Existe muita coisa sobre na Internet, desde outros programas, scritps e tutoriais
- ★ Existem maneiras de melhorar a segurança da sua rede sem fio, seja por filtro de MAC Address, não enviar beacon com nome da rede ou SSID, entre outros,

contudo muitas pessoas não utilizam, seja por não saber ou pelo trabalho de configurar.

Alguns links úteis:

Um pouco sobre Mecanismos de Segurança de Redes IEEE 802.1
Vídeos focados no Wirershark
hack5 (Segurança de modo geral e outras coisas)
Wireshark FAQs
WLAN (IEEE 802.11) capture setup
Segurança Wi-Fi