Labo 3

Joel Schar, Yann Lederrey

Partie 1

- canal utilisé par l'ap : canal 6
- Phase d'initiation :

1	0.000000	Cisco_4d:14:40	RivetNet_d4:c7:31	EAP	63 Request, Identity
2	0.000127	RivetNet_d4:c7:31	Cisco_4d:14:40	EAP	36 Response, Identity
3	0.004110	Cisco_4d:14:40	RivetNet_d4:c7:31	EAP	24 Request, TLS EAP (EAP-TLS)
4	0.004206	RivetNet_d4:c7:31	Cisco_4d:14:40	EAP	24 Response, Legacy Nak (Response Only)
5	0.006529	Cisco_4d:14:40	RivetNet_d4:c7:31	EAP	24 Request, Protected EAP (EAP-PEAP)

  1-2 : Phase d'initiation 3-5 : Négotiation du protocol d'échange

  Dans cette phase (paquet 2) on peut voir l'identité de la personne :

```
▼ Extensible Authentication Protocol
    Code: Response (2)
    Id: 1
    Length: 18
    Type: Identity (1)
    Identity: yann.lederrey
```

- Phase Hello :

6	0.006742	RivetNet_d4:c7:31	Cisco_4d:14:40	TLSv1.2	325 Client Hello
7	0.037246	Cisco_4d:14:40	RivetNet_d4:c7:31	TLSv1.2	1030 Server Hello, Certificate, Server Key Exchang…
8	0.037327	RivetNet_d4:c7:31	Cisco_4d:14:40	EAP	24 Response, Protected EAP (EAP-PEAP)
9	0.040491	Cisco_4d:14:40	RivetNet_d4:c7:31	TLSv1.2	1026 Server Hello, Certificate, Server Key Exchang…
10	0.040548	RivetNet_d4:c7:31	Cisco_4d:14:40	EAP	24 Response, Protected EAP (EAP-PEAP)
11	0.043149	Cisco_4d:14:40	RivetNet_d4:c7:31	TLSv1.2	257 Server Hello, Certificate, Server Key Exchang…

  Version :

```
Version: TLS 1.2 (0x0303)
```

  Méthode de compression proposée par le client :

```
▼ Compression Methods (1 method)
    Compression Method: null (0)
```

  Méthode de compression choisie par le serveur :

```
Compression Method: null (0)
```

  Suites cryptographiques proposée par le client

```
▼ Cipher Suites (85 suites)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
    Cipher Suite: TLS_DH_DSS_WITH_AES_256_GCM_SHA384 (0x00a5)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 (0x00a3)
    Cipher Suite: TLS_DH_RSA_WITH_AES_256_GCM_SHA384 (0x00a1)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x009f)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x006b)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (0x006a)
    Cipher Suite: TLS_DH_RSA_WITH_AES_256_CBC_SHA256 (0x0069)
    Cipher Suite: TLS_DH_DSS_WITH_AES_256_CBC_SHA256 (0x0068)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
    Cipher Suite: TLS_DH_RSA_WITH_AES_256_CBC_SHA (0x0037)
    Cipher Suite: TLS_DH_DSS_WITH_AES_256_CBC_SHA (0x0036)
    Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0088)
    Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA (0x0087)
    Cipher Suite: TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0086)
    Cipher Suite: TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA (0x0085)
    Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384 (0xc032)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02e)
    Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384 (0xc02a)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384 (0xc026)
    Cipher Suite: TLS_ECDH_RSA_WITH_AES_256_CBC_SHA (0xc00f)
    Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA (0xc005)
    Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0084)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Cipher Suite: TLS_DH_DSS_WITH_AES_128_GCM_SHA256 (0x00a4)
    Cipher Suite: TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 (0x00a2)
    Cipher Suite: TLS_DH_RSA_WITH_AES_128_GCM_SHA256 (0x00a0)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x0067)
```

```
Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (0x0040)
Cipher Suite: TLS_DH_RSA_WITH_AES_128_CBC_SHA256 (0x003f)
Cipher Suite: TLS_DH_DSS_WITH_AES_128_CBC_SHA256 (0x003e)
Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
Cipher Suite: TLS_DH_RSA_WITH_AES_128_CBC_SHA (0x0031)
Cipher Suite: TLS_DH_DSS_WITH_AES_128_CBC_SHA (0x0030)
Cipher Suite: TLS_DHE_RSA_WITH_SEED_CBC_SHA (0x009a)
Cipher Suite: TLS_DHE_DSS_WITH_SEED_CBC_SHA (0x0099)
Cipher Suite: TLS_DH_RSA_WITH_SEED_CBC_SHA (0x0098)
Cipher Suite: TLS_DH_DSS_WITH_SEED_CBC_SHA (0x0097)
Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0045)
Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA (0x0044)
Cipher Suite: TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0043)
Cipher Suite: TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA (0x0042)
Cipher Suite: TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 (0xc031)
Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02d)
Cipher Suite: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256 (0xc029)
Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 (0xc025)
Cipher Suite: TLS_ECDH_RSA_WITH_AES_128_CBC_SHA (0xc00e)
Cipher Suite: TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xc004)
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
Cipher Suite: TLS_RSA_WITH_SEED_CBC_SHA (0x0096)
Cipher Suite: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0041)
Cipher Suite: TLS_ECDHE_RSA_WITH_RC4_128_SHA (0xc011)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_RC4_128_SHA (0xc007)
Cipher Suite: TLS_ECDH_RSA_WITH_RC4_128_SHA (0xc00c)
Cipher Suite: TLS_ECDH_ECDSA_WITH_RC4_128_SHA (0xc002)
Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
Cipher Suite: TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA (0xc012)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc008)
Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
Cipher Suite: TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA (0x0010)
Cipher Suite: TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA (0x000d)
Cipher Suite: TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA (0xc00d)
Cipher Suite: TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA (0xc003)
Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
Cipher Suite: TLS_EMPTY_RENEGOTIATION_INFO_SCSV (0x00ff)
```

Cipher suite choisie par le serveur: `Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)`

Nonces :

```
▼ Random: 028c7f2694b67d81ac97d73175cfe0e774f0dc29550e4f2f...
    GMT Unix Time: May 10, 1971 23:20:22.000000000 CET
    Random Bytes: 94b67d81ac97d73175cfe0e774f0dc29550e4f2f3787bf3e...
```

Session ID : `Session ID: e6538179a86c9566205f95d6e569508586bba56bf3366ef9...`

- Phase d'échange de certificat : La transmission du certificat serveur et faites dans la phase Hello.

```
12 0.044423   RivetNet_d4:c7:31   Cisco_4d:14:40   TLSv1.2   150 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
13 0.047665   Cisco_4d:14:40      RivetNet_d4:c7:31   TLSv1.2    75 Change Cipher Spec, Encrypted Handshake Message
14 0.047808   RivetNet_d4:c7:31   Cisco_4d:14:40      EAP        24 Response, Protected EAP (EAP-PEAP)
```

12 : Transmission du Pre Master Secret 13 : Envoie d'une signature au client pour que l'AP s'authentifie auprès du client (hash de touts les échanges) 14 : Ack

Certificat serveur : Change cipher spec :

```
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
```

```
0000  9c b6 d0 d4 c7 31 dc a5  f4 4d 14 40 88 8e 02 00   ·····1·· ·M·@····
0010  00 39 01 28 00 39 19 01  14 03 03 00 01 01 16 03   ·9·(·9·· ·····:··
0020  03 00 28 32 07 0e 18 e5  87 a3 5a a9 62 a5 e4 88   ··(2···· ··Z·b···
0030  5d 89 ea db 62 cb ca 4f  99 c9 e0 5c ac 1c 21 2f   ]···b··O ···\··!/
0040  3b 0c 68 52 88 ae 19 9d  c8 84 9a                 ;·hR···· ···
```

- Authentification interne et transmission de la clé WPA (échange chiffré) :

```
15 0.050303    Cisco_4d:14:40    RivetNet_d4:c7:31    TLSv1.2    58 Application Data
16 0.050372    RivetNet_d4:c7:31 Cisco_4d:14:40       TLSv1.2    71 Application Data
17 0.052792    Cisco_4d:14:40    RivetNet_d4:c7:31    TLSv1.2    84 Application Data
18 0.052942    RivetNet_d4:c7:31 Cisco_4d:14:40       TLSv1.2   125 Application Data
19 0.058811    Cisco_4d:14:40    RivetNet_d4:c7:31    TLSv1.2   104 Application Data
20 0.058924    RivetNet_d4:c7:31 Cisco_4d:14:40       TLSv1.2    59 Application Data
21 0.061281    Cisco_4d:14:40    RivetNet_d4:c7:31    TLSv1.2    57 Application Data
```

- Success de l'échange:

```
22 0.061352    RivetNet_d4:c7:31    Cisco_4d:14:40    EAP    24 Response, Protected EAP (EAP-PEAP)
23 0.069797    Cisco_4d:14:40       RivetNet_d4:c7:31 EAP    22 Success
```

- 4-way handshake:
-

```
24 0.069963    Cisco_4d:14:40    RivetNet_d4:c7:31    EAPOL    135 Key (Message 1 of 4)
25 0.070179    RivetNet_d4:c7:31 Cisco_4d:14:40       EAPOL    135 Key (Message 2 of 4)
26 0.071587    Cisco_4d:14:40    RivetNet_d4:c7:31    EAPOL    169 Key (Message 3 of 4)
27 0.071647    RivetNet_d4:c7:31 Cisco_4d:14:40       EAPOL    113 Key (Message 4 of 4)
```

> **Question :_** Quelle ou quelles méthode(s) d'authentification est/sont proposé(s) au client ?
>
> ***Réponse :*** Le serveur propose d'abord "EAP-TLS", le client refuse, le serveur propose ensuite "EAP-PEAP"
>
> ```
> 3 0.004110    Cisco_4d:14:40    RivetNet_d4:c7:31    EAP    24 Request, TLS EAP (EAP-TLS)
> 4 0.004206    RivetNet_d4:c7:31 Cisco_4d:14:40       EAP    24 Response, Legacy Nak (Response Only)
> 5 0.006529    Cisco_4d:14:40    RivetNet_d4:c7:31    EAP    24 Request, Protected EAP (EAP-PEAP)
> ```

> ***Question:*** Quelle méthode d'authentification est utilisée ?
>
> ***Réponse:*** "EAP-PEAP"

> ***Question:*** Lors de l'échange de certificats entre le serveur d'authentification et le client :
>
> - Le serveur envoie-t-il un certificat au client ? Pourquoi oui ou non ?
>
> ***Réponse:*** Oui le serveur envoie son certificat au client afin de s'authentifier. (c'est demandé par le protocol "EAP-PEAP")
>
> - b. Le client envoie-t-il un certificat au serveur ? Pourquoi oui ou non ?
>
> ***Réponse:*** Non, dans le cas de "EAP-PEAP", le client n'envoie pas de certificat (contrairement au protocol "EAP-TLS", dans lequel les deux envoient un certificat.)

Partie 2

Dans le fichier `/etc/hostapd-wpe/hostapd-wpe.conf` il faut choisir le SSID que l'on veut générer le canal

```
interface=wlan0mon
...
# 802.11 Options
ssid=HEIG-VD-GUEST
channel=6
```

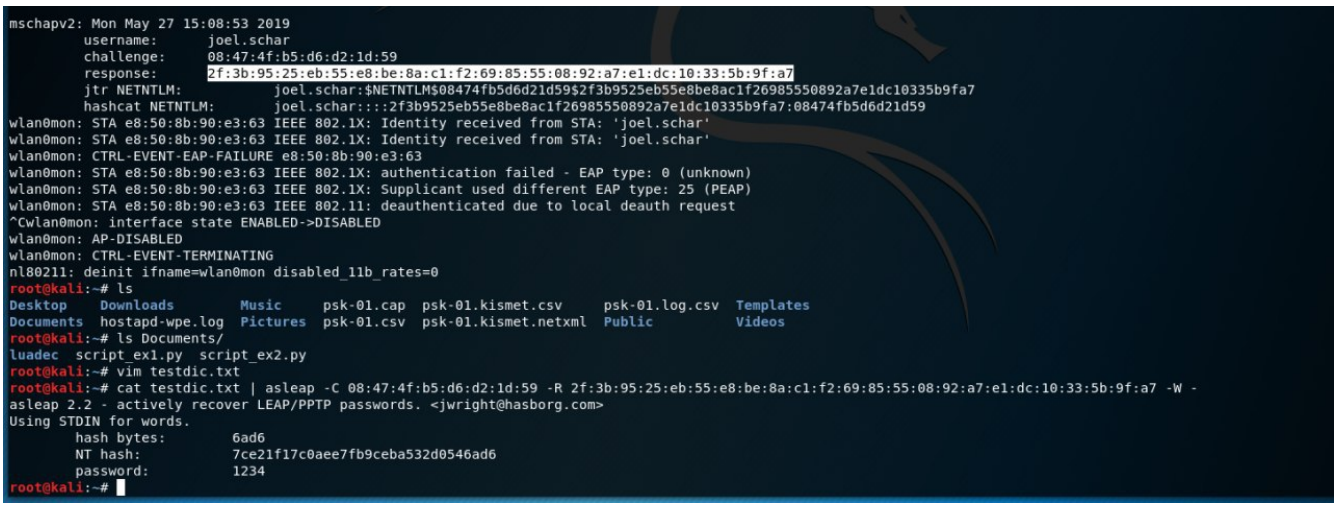
Il faut mettre l'interface en mode monitor

On peut ensuite starter le tool qui va générer un AP avec le ssid que l'on a choisi ici HEIG-VD-GUEST.

En se connectant dessus il catch le challenge response échanger avec l'AP. Depuis celui-ci il est possible de faire un brut force sur le mot de passe utilisé avec l'outil `asleap` qui permet de faire le calcul challenge - responce et faire la comparaison.

```
mschapv2: Mon May 27 15:08:44 2019
     username:  joel.schar
     challenge: f3:a9:bd:d3:09:53:b0:3b
     response:  ba:ed:ae:32:ff:57:ba:88:32:0d:e4:ef:a8:bb:c4:04:4f:29:f7:4c:41:26:31:f5
     jtr NETNTLM:
joel.schar:$NETNTLM$f3a9bdd30953b03b$baedae32ff57ba88320de4efa8bbc4044f29f74c412631f5
     hashcat NETNTLM:
joel.schar::::baedae32ff57ba88320de4efa8bbc4044f29f74c412631f5:f3a9bdd30953b03b
```

Avec un dictionnaire custom contenant le mot de passe que l'on inject dans asleap on peut récupérer le mot de passe utiliser pour calculer le challenge et donc le mot de passe de l'utilisateur.



> **Question :** Quelles modifications sont nécessaires dans la configuration de hostapd-wpe pour cette attaque ?

> **Réponse :** définir le ssid et donner le channel, ici : HEIG-VD-GUEST, channel 6, et indiquer l'interface utilisée.

> **Question:** Quel type de hash doit-on indiquer à john pour craquer le handshake ?
>
> **Réponse:** On indique le challenge et le response.

> **Question:** 6.Quelles méthodes d’authentification sont supportées par hostapd-wpe ?
>
> **Réponse:** il support :
>
> - EAP-FAST/MSCHAPv2
> - PEAP/MSCHAPv2
> - EAP-TTLS/MSCHAPv2
> - EAP-TTLS/MSCHAP
> - EAP-TTLS/CHAP
> - EAP-TTLS/PAP