IUT de La Rochelle

Université La Rochelle

- TP 3 -
Wi-Fi et RADIUS

par Édouard Lumet

IT La Rochelle

iut réseaux & télécoms

Sommaire

1. Configuration serveur RADIUS

Comme pour le TP1, nous travaillons sous Fedora. La configuration du serveur RADIUS se fait dans le répertoire */etc/raddb*. Les fichiers de configuration notables sont :

- *radiusd.conf* : fichier de configuration principal,
- *users* : fichier déclarant les utilisateurs locaux avec leur méthode d'authentification auprès du RADIUS,
- *clients.conf* : on déclare dans ce fichier le (ou les) client(s) pouvant se connecter au RADIUS (notre AP entre autres),
- *eap.conf* : définit la méthode à employer pour s'identifier auprès du RADIUS (leap, tls, …).

Ici, nous allons commencer par modifier le fichier *eap.conf*. Dans la partie **eap{}**, on affecte la valeur **leap** à la variable **default_eap_type**.

Ensuite, on ajoute un accès utilisateur dans le fichier *users* :

```
admin           Auth-Type := EAP, User-Password := "rt2ATP"
                Reply-Message = "Authentification de %u"
```

On a ainsi ajouté un accès pour l'utilisateur **admin** avec le mot de passe **rt2ATP**. Le type d'authentification est **EAP**.

On ajoute enfin un utilisateur via le fichier *clients.conf* :

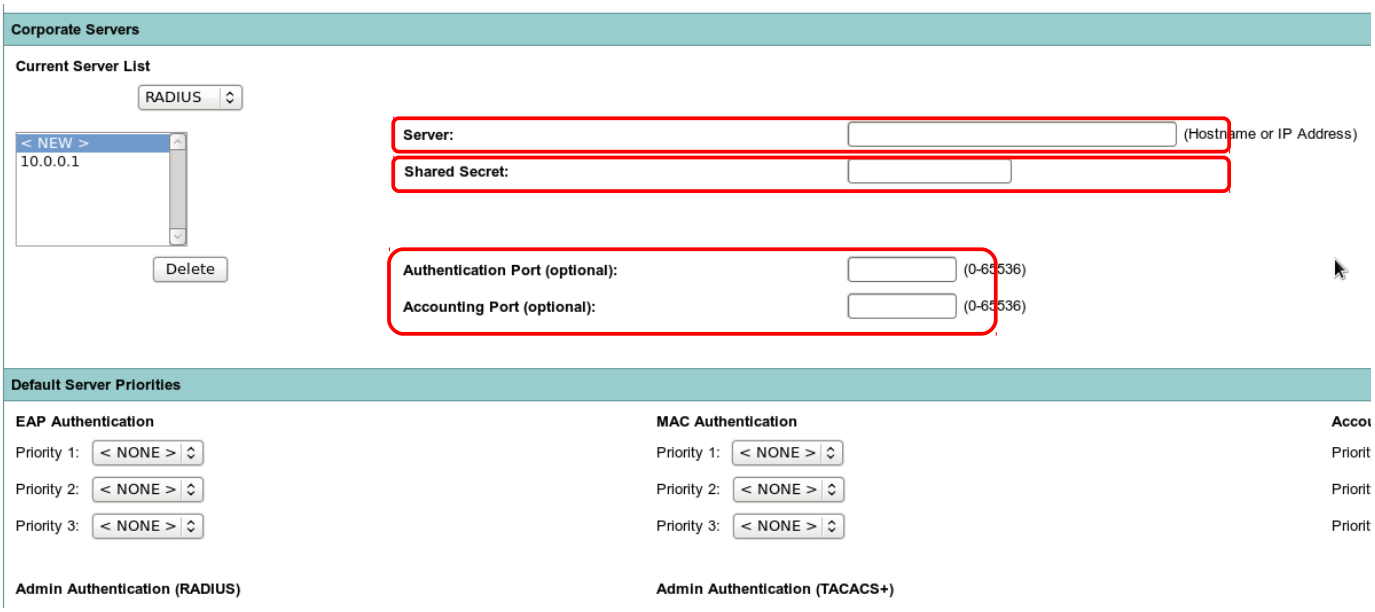
```
client 10.0.0.10/24 {
        secret = wifiradius
        shortname = ap
        nastype = cisco
}
```

La machine cliente **10.0.0.1** aura alors accès au RADIUS en utilisant le secret **wifiradius**. À noter que le type de NAS est **cisco**. Cela permet donc à l'AP de communiquer avec le serveur RADIUS.

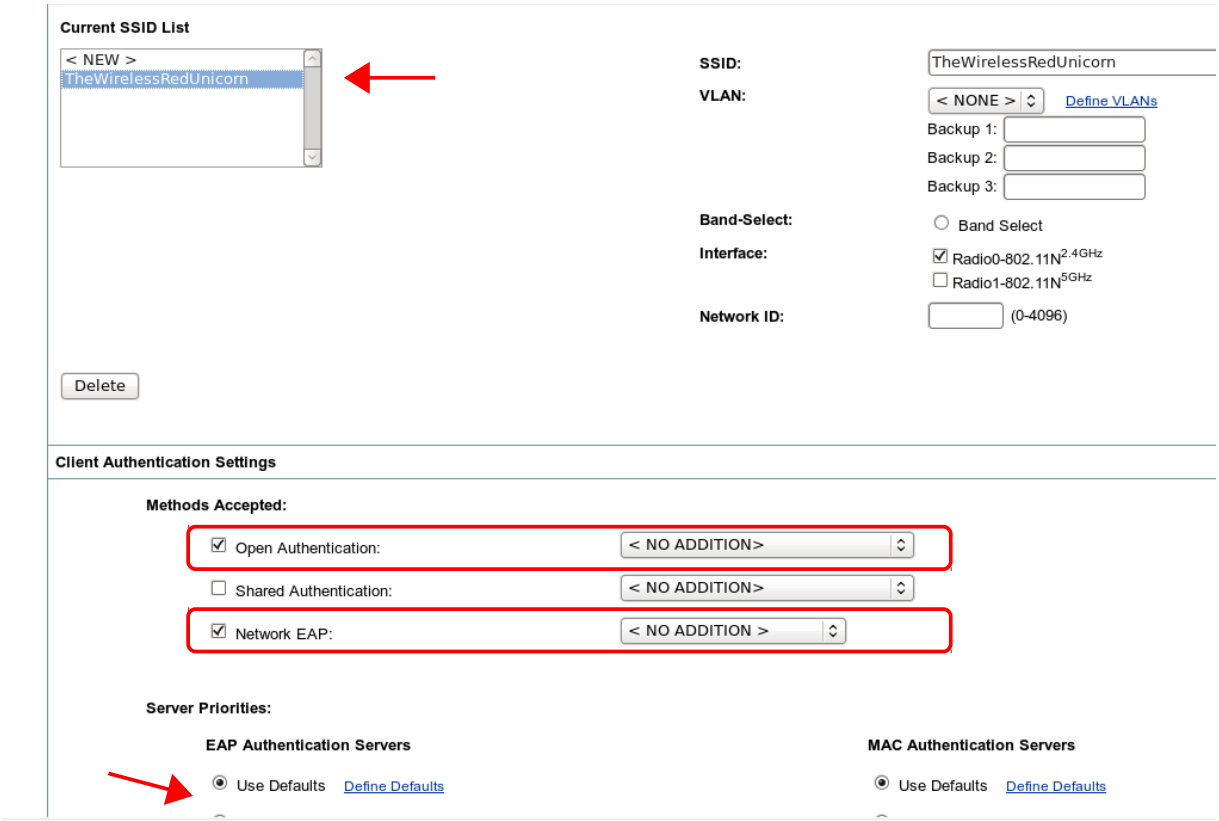
2. Configuration authenticator

L'adresse IP indiquée précédemment (dans le fichier *clients.conf*) est déjà celle attribuée à l'interface Ethernet de l'AP, par le serveur DHCP installé sur le PC.

Dans l'interface graphique de l'AP, on effectue quelques configurations afin de lui indiquer qu'il doit communiquer avec le RADIUS et comment il doit communiquer avec ce dernier.



On se trouve ici dans SECURITY > Server Manager. Dans le champ *Server* on indique l'adresse IP du RADIUS (notre PC), soit 10.0.0.1. Dans *Shared Secret*, on indique le secret partagé pour s'authentifier auprès du serveur. Cet authentification est basée sur un échange CHAP entre l'AP et le RADIUS. Le sercet est hashé par chacune des deux machines puis le hash est comparé afin de vérifier si le secret est le même. Pour finir, on indique les ports *1812* et *1813* pour, respectivement, *Authentication Port* et *Accounting Port*.



Ensuite, dans la section SECURITY > SSID Manager on modifie la méthode d'authentification auprès de l'AP. Une fois le SSID sélectionné, on coche *Open Authentification* puis on sélectionne dans la liste *With EAP*. On coche également *Network EAP*. Plus bas, sous Server Priorities, EAP Authentication Servers, on coche *Customize* et on sélectionne dans la liste notre RADIUS *10.0.0.1*.

Pour finir, on définit une méthode de chiffrement dans SECURITY > Encryption Manager, comme nous avons fait lors du TP1. On choisit par exemple *WEP (40 bits)* dans le menu déroulant *Cipher* puis en dessous, on saisit une clé de 10 caractères (ou 26 pour une clé de 128 bits). Ici, j'ai choisi la clé *0123456789* pour constituer la clé WEP.

3. Configuration du supplicant

La configuration du supplicant donnée dans l'énoncé est pour Linux Mint. Sous Fedora par exemple, il nous est plus simple d'utiliser le Network Manager afin de se connecter à notre réseau Wi-Fi et d'observer les échanges entre le supplicant (client) et l'AP puis entre l'AP et le serveur RADIUS.

À l'aide de mon ordinateur personnel (sous Ubuntu), je me suis connecté à mon réseau Wi-Fi après avoir démarré Wireshark sur mon PC (interface WLAN) et sur le PC de la salle (interface ETH1). La première capture Wireshark permet de visualiser les échanges entre le client ou supplicant et l'AP. La seconde permet d'observer les échanges entre l'AP et le serveur RADIUS lors de l'authentification.

Échanges Supplicant ↔ AP

Les échanges entre le client et l'AP sont les suivants :

8	AP	→	PC (supplicant)	EAP	Request, Identity
9	PC (supplicant)	→	AP	EAP	Response, Identity
10	AP	→	PC (supplicant)	EAP	Request, Cisco Wireless EAP / Lightweight EAP (EAP-LEAP)
11	PC (supplicant)	→	AP	EAP	Response, Cisco Wireless EAP / Lightweight EAP (EAP-LEAP)
12	AP	→	PC (supplicant)	EAP	Success
13	PC (supplicant)	→	AP	EAP	Request, Cisco Wireless EAP / Lightweight EAP (EAP-LEAP)
14	AP	→	PC (supplicant)	EAP	Response, Cisco Wireless EAP / Lightweight EAP (EAP-LEAP)
15	AP	→	PC (supplicant)	EAPOL	Key
16	AP	→	PC (supplicant)	EAPOL	Key

```
802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: EAP Packet (0)
  Length: 10
▾ Extensible Authentication Protocol
    Code: Response (2)
    Id: 1
    Length: 10
    Type: Identity (1)
    Identity: admin
```

La réponse du PC à l'AP (trame 9, *EAP Identity Repsonse*) contient l'identité requise par le PC pour la connexion, soit *admin*.

Ensuite, l'AP envoie un message EAP (*LEAP*) *Request* au client pour lui demander les informations d'authentification pour l'utilisateur admin. Le client envoie donc le mot de passe hashé à l'AP en réponse.

```
802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: EAP Packet (0)
  Length: 37
▾ Extensible Authentication Protocol
    Code: Response (2)
    Id: 2
    Length: 37
  ▸ Type: Cisco Wireless EAP / Lightweight EAP (EAP-LEAP) (17)
    EAP-LEAP Version: 1
    EAP-LEAP Reserved: 0x00
    EAP-LEAP Count: 24
    EAP-LEAP Peer-Response: 1b95817f9ee92a949e88c19725a5af40b4e7c7d219bf0e5e
    EAP-LEAP Name: admin
```

À partir de ce message, des échanges décrits ci-dessous s'effectuent entre l'AP et le serveur RADIUS pour confirmer les identifiants donnés par le client pour *admin*.

Lorsque le RADIUS a confirmé l'authentification à l'AP, ce dernier envoie alors un message *EAP Success* au client lui indiquant qu'il est correctement authentifié.

```
802.1X Authentication
    Version: 802.1X-2001 (1)
    Type: EAP Packet (0)
    Length: 4
▾ Extensible Authentication Protocol
    Code: Success (3)
    Id: 3
    Length: 4
```

```
15 CiscoInc_… Azurewav_9… EAPOL   Key
16 CiscoInc_… Azurewav_9… EAPOL   Key

Frame 15: 67 bytes on wire (536 bits), 67 bytes captu
Ethernet II, Src: CiscoInc_77:5e:20 (0c:27:24:77:5e:2
▸ Destination: Azurewav_9d:c6:d8 (54:27:1e:9d:c6:d8)
▸ Source: CiscoInc_77:5e:20 (0c:27:24:77:5e:20)
  Type: 802.1X Authentication (0x888e)
802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 49
  Key Descriptor Type: RC4 Descriptor (1)
  Key Length: 5
  Replay Counter: 47903192711171
  Key IV: 805b78bb052ff03624d211c622365077
▾ Key Index: 0x00
    0... .... = Type: Broadcast
    .000 0000 = Number: 0
  Key Signature: eb9022219aff1f32b65e40bb612299e7
  Key: 5f0a8008dc
  Key Generated Locally: False
```

Dans les messages *EAPOL Key* suivants, de l'AP vers le client, on peut voir l'algorithme de chiffrement utilisé. Celui-ci est RC4, un algorithme de chiffrement symétrique faible et « facilement » réversible.

Le vecteur d'initialisation est également donné : 805b78bb052ff03624d211c622365077. C'est ce vecteur qui est utilisé lors du chiffrement avec WEP. Il est commun à tous et est permanent. Ici, l'AP en fournit deux, un de type *broadcast* et un autre de type *unicast*. EAPOL (EAPoL) désigne EAP over LAN.

```
16 CiscoInc_… Azurewav_9… EAPOL   Key

Frame 16: 62 bytes on wire (496 bits), 62 bytes captu
Ethernet II, Src: CiscoInc_77:5e:20 (0c:27:24:77:5e:2
▸ Destination: Azurewav_9d:c6:d8 (54:27:1e:9d:c6:d8)
▸ Source: CiscoInc_77:5e:20 (0c:27:24:77:5e:20)
  Type: 802.1X Authentication (0x888e)
802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 44
  Key Descriptor Type: RC4 Descriptor (1)
  Key Length: 5
  Replay Counter: 47903192711172
  Key IV: 606f7228bf20114a5e207d41d5c670f7
▾ Key Index: 0x83
    1... .... = Type: Unicast
    .000 0011 = Number: 3
  Key Signature: 21c5fbef5eb33ab2898b5bf8488b9398
  Key Generated Locally: True
```

Échanges AP ↔ serveur RADIUS

Les échanges entre l'AP et le serveur RADIUS lors de la connexion et l'authentification d'un client au réseau Wi-Fi sont les suivants :

```
21  AP          → Srv RADIUS   RADIUS  Access-Request(1) (id=11, l=198)
22  Srv RADIUS  → AP           RADIUS  Access-Challenge(11) (id=11, l=86)
23  AP          → Srv RADIUS   RADIUS  Access-Request(1) (id=12, l=182)
24  Srv RADIUS  → AP           RADIUS  Access-Accept(2) (id=12, l=167)
```

Le premier message provient de l'AP. C'est un *Access-Request* qui indique au serveur RADIUS qu'une station (*Calling-Station 5427.1e9d.c6d8*, mon PC) tente de s'authentifier auprès de l'AP via le réseau Wi-Fi TheWirelessRedUnicorn (*Called-Station : 0c-27-24-77-5E-20:TheWirelessRedUnicorn*). Le nom d'utilisateur saisi par le client est *admin*.

```
RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0xb (11)
  Length: 198
  Authenticator: 6ec4782063fb21865d44d39b642e9476
  [The response to this request is in frame 22]
▾ Attribute Value Pairs
  ▸ AVP: l=7 t=User-Name(1): admin
  ▸ AVP: l=6 t=Framed-MTU(12): 1400
  ▸ AVP: l=41 t=Called-Station-Id(30): 0C-27-24-77-5E-20:TheWirelessRedUnicorn
  ▸ AVP: l=16 t=Calling-Station-Id(31): 5427.1e9d.c6d8
  ▸ AVP: l=6 t=Service-Type(6): Login(1)
  ▸ AVP: l=18 t=Message-Authenticator(80): d3419cdf7858a5f3916feea2fff341e2
  ▸ AVP: l=39 t=EAP-Message(79) Last Segment[1]
  ▸ AVP: l=6 t=NAS-Port-Type(61): Wireless-802.11(19)
  ▸ AVP: l=6 t=NAS-Port(5): 263
  ▸ AVP: l=5 t=NAS-Port-Id(87): 263
  ▸ AVP: l=18 t=State(24): e4bd4ed6e4bf5f43069bdff6e86f6bde
  ▸ AVP: l=6 t=NAS-IP-Address(4): 10.0.0.10
  ▸ AVP: l=4 t=NAS-Identifier(32): ap
```

Ensuite, le RADIUS répond par un message de type *Access-Challenge*. Ce message permet de demander au client RADIUS (ici, l'AP) plus d'informations afin de poursuivre l'authentification.

```
  Code: Access-Challenge (11)
  Packet identifier: 0xb (11)
  Length: 86
  Authenticator: 108768c4d0f87eff2904bd05b26a4acc
  [This is a response to a request in frame 21]
  [Time from request: 0.000215000 seconds]
▾ Attribute Value Pairs
  ▸ AVP: l=24 t=Reply-Message(18): Authentification de %u
  ▸ AVP: l=6 t=EAP-Message(79) Last Segment[1]
  ▸ AVP: l=18 t=Message-Authenticator(80): 234ca987c4cfa50951d6c4394c9f5e9f
  ▸ AVP: l=18 t=State(24): e4bd4ed6e5be5f43069bdff6e86f6bde
```

Dans un troisième temps, l'AP répond par un nouvel *Access-request* avec des informations complémentaires. On voit notamment dans ce message le type d'authentification qui est *LEAP*, ce qui n'apparaissait pas dans le premier message de la part de l'AP.

```
RADIUS Protocol
  Code: Access-Request (1)
  Packet identifier: 0xc (12)
  Length: 182
  Authenticator: b174104e93b30cf8d01c525af241012c
  [The response to this request is in frame 24]
▾ Attribute Value Pairs
  ▸ AVP: l=7 t=User-Name(1): admin
  ▸ AVP: l=6 t=Framed-MTU(12): 1400
  ▸ AVP: l=41 t=Called-Station-Id(30): 0C-27-24-77-5E-20:TheWirelessRedUnicorn
  ▸ AVP: l=16 t=Calling-Station-Id(31): 5427.1e9d.c6d8
  ▸ AVP: l=6 t=Service-Type(6): Login(1)
  ▸ AVP: l=18 t=Message-Authenticator(80): 8ef95cb5b0500a9a06a46a5c65b373e0
  ▾ AVP: l=23 t=EAP-Message(79) Last Segment[1]
      AVP Type: 79
      AVP Length: 23
      EAP fragment: 01030015110100083641ed5d6ab8ba4961646d696e
    ▾ Extensible Authentication Protocol
        Code: Request (1)
        Id: 3
        Length: 21
      ▸ Type: Cisco Wireless EAP / Lightweight EAP (EAP-LEAP) (17)
        EAP-LEAP Version: 1
        EAP-LEAP Reserved: 0x00
        EAP-LEAP Count: 8
        EAP-LEAP Data: 3641ed5d6ab8ba49
        EAP-LEAP Name: admin
  ▸ AVP: l=6 t=NAS-Port-Type(61): Wireless-802.11(19)
  ▸ AVP: l=6 t=NAS-Port(5): 263
  ▸ AVP: l=5 t=NAS-Port-Id(87): 263
  ▸ AVP: l=18 t=State(24): e4bd4ed6e5be5f43069bdff6e86f6bde
  ▸ AVP: l=6 t=NAS-IP-Address(4): 10.0.0.10
  ▸ AVP: l=4 t=NAS-Identifier(32): ap
```

Le message EAP contient lui aussi les informations concernant le nom d'utilisateur. Un champ en surbrillance attire l'attention sur le fait que LEAP est désormais à éviter car une simple attaque par dictionnaires peut compromettre la sécurité de notre réseau. De plus, cette réponse de l'AP contient de plus amples informations sur l'AP lui même, de type *Cisco*, d'adresse *10.0.0.10*, de nom *ap*.

Enfin, le message *Access-Accept* de la part du serveur RADIUS indique que l'authentification est permise pour le client (ici, mon PC) auprès de l'AP.

```
RADIUS Protocol
  Code: Access-Accept (2)
  Packet identifier: 0xc (12)
  Length: 167
  Authenticator: 7af19cdb22ae7025a57b2bc0b0ffaa5f
  [This is a response to a request in frame 23]
  [Time from request: 0.000224000 seconds]
▾ Attribute Value Pairs
  ▸ AVP: l=24 t=Reply-Message(18): Authentification de %u
  ▸ AVP: l=59 t=Vendor-Specific(26) v=ciscoSystems(9)
  ▸ AVP: l=39 t=EAP-Message(79) Last Segment[1]
  ▸ AVP: l=18 t=Message-Authenticator(80): 3c9320bc5b13afb01139ef56d52fcf1e
  ▸ AVP: l=7 t=User-Name(1): admin
```

C'est suite à ce message que l'AP va envoyer un message EAP de type *Success* vers le client pour lui indiquer que l'authentification est correcte.

Ces échanges RAIDUS s'inscrivent donc entre les messages *EAP response* et *EAP success* vus plus haut lors des échanges entre le client et l'AP.