RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ D'ABOMEY-CALAVI**

**INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE**

BP 526 Cotonou Tel : +229 21 14 19 88
https://www.ifri-uac.net Courriel : contact@ifri.uac.bj

# MÉMOIRE

pour l'obtention du

**Diplôme de Licence en Informatique**

**Option :** Sécurité Informatique

**Présenté par** :
Bleck Steeven **KOCOUVISSO**

# Mise en place d'un système d'authentification basé sur EAP-TLS dans un réseau wifi

**Sous la supervision** :
Ing Francky **ADEOSSI**

**Membres du jury** :

| | |
|---|---|
| Charles SOBABE | Dr : Président |
| Pascal FOLOKOUE | Ing : Examinateur |
| Francky ADEOSSI | Ing : Rapporteur |

**date de soutenance** : 14 décembre 2022

Année Académique : **2021-2022**

# Sommaire


Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Liste des figures vi

Liste des acronymes viii

Introduction 1

1 Revue Littéraire 3

2 Analyse choix technique et Conception 12

3 Résultat obtenu et Perspectives 22

Conclusion 33

References 34

Webographie 35

Table des matières 37

# Dédicace

A
Mon père KOCOUVISSO K. Gaston
Ma mère CAKPO T. Francisca
feu ma grand-mère HOUNDETOUNKPO Solange

# Remerciements

Nous exprimons notre profonde gratitude à l'Éternel, Dieu Créateur, pour son amour infini, pour l'inspiration et la protection dont il nous comble tous les jours. À travers ce mémoire, nous tenons à remercier :

- Le Professeur Eugène C. EZIN, le Directeur de notre institut pour nous avoir assuré cette formation ;
- M. GUEDEGBE Fabrice, Chef adjoint du service réseau et maintenance de la DGTCP, notre superviseur pour avoir accepté, diriger et suivre avec grand intérêt le déroulement de ce projet.
- Monsieur Francky ADEOSSI, notre maître de mémoire, pour ses sages conseils et ses orientations ;
- Tous les enseignants de notre institut, en particulier ceux de la filière Sécurité Informatique pour avoir accepté partager une partie de leurs connaissances avec nous ;
- Tous les étudiants de la cinquième promotion de Génie Logiciel/Sécurité Informatique/Internet et Multimédia de l'Institut de Formation et de Recherche en Informatique(IFRI) ;
- Tous ceux qui m'ont aidé à la réalisation de ce mémoire.

# Résumé

La façon la plus simple de connecter deux équipements réseau, c'est d'utiliser un câble. Mais il est parfois plus commode de pouvoir s'en affranchir. La technologie Wi-Fi, depuis sa première utilisation, tient la promesse d'un « Ethernet sans fil » faisant de lui l'un des réseaux les plus importants du monde informatique. Néanmoins, il reste vulnérable par certains aspects de sa conception, sa visibilité, mais aussi au développement d'attaques et à l'existence de vulnérabilités connues.
Le présent travail de mémoire met donc en place une solution contre ces dits attaques, c'est-à-dire une solution pour sécuriser un réseau sans fil afin d'y restreindre l'accès.
La mise en place et l'implémentation de cette solution de sécurisation a été rendu possible grâce aux travaux antérieurs sur la sécurité des réseaux sans fil . La solution mise en place fournit donc relativement de bons résultats et des améliorations à la sécurité du réseau.

**Mots clés** : Wi-Fi, réseau sans fil, sécuriser, authentifier.

# Abstract

The easiest way to connect two network devices is to use a cable. But sometimes it is more convenient to be able to get rid of it. Wi-Fi technology; since its first use, keeps the promise of "wireless Ethernet" making it one of the most important networks in the computer world. Nevertheless, it remains vulnerable by certain aspects of its design, its visibility but also to the development of attacks and the existence of known vulnerabilities.
This dissertation therefore puts in place a solution against these attacks, i.e. a solution to secure a wireless network in order to restrict access to it.
The setting up and implementation of this security solution has been made possible thanks to earlier work on wireless network security. The solution implemented therefore provides relatively good results. and improvements to network security.

**Keywords**: Wi-Fi, wireless network, secure, authenticate.

# Liste des figures


1.1 Classification des réseaux sans fil . . . 5
1.2 Les différentes normes Wi-Fi . . . 7
1.3 Adaptateur USB . . . 8
1.4 Carte wifi PCI . . . 8
1.5 Un point d'accès sans fil . . . 8
1.6 Modes infrastructure et Ad-hoc . . . 9

2.1 Les 3 entités qui interagissent dans le protocole 802.1x . . . 13
2.2 Les trafics EAP . . . 14
2.3 Les étapes d'une connexion EAP . . . 15
2.4 Etude comparative de quelques types du protocole d'authentification EAP . . . 17
2.5 Etapes d'une authentification EAP-TLS . . . 19

3.1 Contenu du package nécessaire à l'installation de FreeRadius . . . 22
3.2 Exécution du fichier install de notre package d'installation de FreeRadius . . . 23
3.3 Version de freeradius installée . . . 23
3.4 lancement de FreeRadius . . . 24
3.5 Début de la génération du certificat racine . . . 24
3.6 Génération du certificat racine . . . 25
3.7 Convertion du certificat racine dans un fichier binaire . . . 25
3.8 Début de génération du certificat serveur . . . 25
3.9 Génération du certificat serveur . . . 26
3.10 Confirmation . . . 26
3.11 Début de génération du certificat client . . . 27
3.12 Génération du certificat client . . . 27
3.13 Confirmation . . . 27
3.14 Génération des paramètres de l'algorithme de Diffie-Hellman . . . 28
3.15 Passage à eap-tls . . . 28
3.16 Implémentation du path du certificat serveur . . . 28
3.17 Implémentation du path du certificat racine . . . 28
3.18 Etat par défaut des valeurs de génération des fichiers aléatoires d'échange de clés . . . 29
3.19 Configuration des fichiers aléatoires d'échange de clés . . . 29
3.20 Valeur d'authentification du client par défaut . . . 29
3.21 Valeur d'authentification du client décommenter . . . 29
3.22 Clients . . . 30
3.23 Information du point d'accès . . . 30
3.24 Téléchargement du pacquage eapol-test . . . 31

3.25 Configuration de eapol-test . . . . . 31

# Liste des acronymes

**802.1x :**

Port Based Network Access Control

**AAA :**

Authentication Authorization Accounting

**ADSL :**

Asymmetric Digital Subscriber Line

**ASCII :**

American Standard Code for Information Interchange

**BSS :**

Basic service set

**BSSID :**

Basic Set Service Identifier

**EAP :**

Extensible Authentication Protocol

**EAP :**

Wireless Personal area network

**FAST :**

Flexible Authentication via Secure Tunneling

**GPRS :**

General Packet Radio Service

**GSM :**

Global System for Mobile Communications

**hiperLAN2 :**

Igh Performance Radio LAN 2.0

**IBSS :**

Independent Basic Service Set

**IEEE :**

Institute of Electrical and Electronics Engineers

**LEAP :**

Lightweight Extensible Authentication Protocol

**MAC :**

Media Access Control

**MD :**

Message Digest 5

**NAC :**

Network Access Server

**OSI :**

Open Systems Interconnection

**PAE :**

Port Access Entity

**PAP :**

Password Authentication Protocol

**PCI :**

Peripheral Component Interconnect

**PCMCIA :**

Personal Computer Memory Card International Association

**PDA :**

Personal Digital Assistant

**PEAP :**

Protected Extensible Authentication Protocol

**SIM :**

Subscriber Identification module

**SSL :**

Secure Socket Layer

**TACACS :**

Terminal Access Controller Access-Control System

**TLS :**

Transport Layer Security

**TSI :**

European Tele- communications Standards Institute

**TTLS :**

Tunneled Transport Layer Security

**UMTS :**

Universal Mobile Telecommunications System

**USB :**

Universal Serial Bus

**WECA :**

Wireless Ethernet Compatibility Alliance

**WEP :**

Wired Equivalent Privacy

**WiFi :**

Wireless Fidelity

**WLAN :**

Wireless local area network

**WMAN :**

Wireless metropolitan area network

# Introduction Générale

L'échange d'informations et de données entre les matériels et équipements réseau n'a lieu que suite à l'effectivité d'une liaison préétablie entre ces équipements. Pendant bien longtemps, cette connexion se faisait physiquement au moyen de câbles réseau (fil) divers et variés ; mais l'élaboration du standard IEEE 802.11 et son développement rapide a été un pas important dans l'avènement et l'évolution des réseaux sans fil, que ce soit en entreprise ou chez les particuliers. Les connexions pouvant ainsi désormais se faire sans l'utilisation de fil, ce qui apporte plus de confort tant dans la mobilité, la flexibilité et l'évolution de l'architecture réseau.

## Problématique

Bien qu'offrant de nombreux avantages, le plus grand enjeu des réseaux sans fil est la sécurité de l'accès et des échanges dans de tels réseaux. En effet, contrairement aux réseaux filaires, la connexion à un réseau Wi-Fi ne nécessite pas d'accéder au matériel physique ; il suffit de rester à bonne distance des ondes émises, ce qui rend d'autant plus difficile le contrôle de l'identité des ordinateurs et des équipements se connectant à notre réseau. À cela, il conviendrait d'ajouter le développement de formes nouvelles d'attaques visant à frauduleusement accéder au réseau. Quels sont les attaques auxquelles est exposé un réseau sans fil Wi-Fi ? Existe-t-il des méthodes ou techniques permettant d'assurer leur sécurité ? Si oui, qu'elles sont-elles ? Autant de questions et préoccupations soulevées auxquels le présent travail s'efforcera de proposer des solutions.

## Objectif

C'est dans cette optique que nous avons abordé notre projet dont l'objectif principal sera de sécuriser un réseau sans fil en se basant sur la solution EAP-TLS afin de garentir la confidentialité, la disponibilité et l'intégrité du réseau. De façon plus spécifique, nous devrons :

- Faire une étude des réseaux Wi-Fi et leur fonctionnement ;
- Parcourir les méthodes et techniques de sécurisation des réseaux sans fil et en faire ressortir la meilleure proposition ;
- Mettre en place un serveur Radius et y implémenter les protocoles de sécurisation nécessaires.

## Organisation du mémoire

Pour mener à bien ce travail, trois chapitres ont été abordés. Le premier chapitre passera en revue les thèmes phares et concepts clés liés aux réseaux sans fil Wi-Fi. Le deuxième, quant à lui, porte sur le procédé d'analyse et de conception utilisé ainsi que sur les choix techniques réalisés pour mener à bien la tâche. Dans le troisième chapitre, est présenté notre solution tout en mettant en avant les résultats issus de nos simulations après les avoir exposés. Enfin, une conclusion qui synthétisera notre travail et présentera les perspectives envisagées.

Chapitre 1

# Revue Littéraire

ce chapitre présente

## Introduction

Le domaine de la communication a connu un essor très important depuis la création d'internet. Cette évolution est principalement due à l'avènement des réseaux sans fil. Ce chapitre présente les réseaux sans fil, leur fonctionnement ainsi que quelques notions importantes qui y sont liées.

## 1.1 Généralité sur les réseaux sans fil

### 1.1.1 Historique et Définition

#### 1.1.1.1 Historique

En 1997, alors que l'attention est accaparée par le succès d'internet et l'euphorie boursière montante, un événement est passé inaperçu sauf pour quelques spécialistes et observateurs : l'adoption du standard IEEE 802.11 ou Ethernet sans fil. Exploitant la bande de fréquence de 2,4 GHz, le 802.11 plafonne à un débit de 2 Mbit/s au maximum. Ce précurseur est suivi de plusieurs déclinaisons dont le Wi-Fi. Il désigne les différentes déclinaisons de la norme IEEE 802.11 qui permet à plusieurs ordinateurs de communiquer sans fil en utilisant comme support les ondes radio. Les câbles disparaissent enfin. Le Wi-Fi est une technologie intéressante pour de nombreuses sociétés liées au monde des télécoms et d'Internet. Les collectivités locales et surtout les particuliers profitent de la facilité d'accès à Internet haut débit liée à cette norme. Dans sa déclinaison la plus connue, 802.11 b, le Wi-Fi utilise la bande de fréquence de 2,4 GHz et atteint un débit théorique de 11 Mbit/s (contre 128, 512 Kbit/s ou 1 Mbit/s pour l'ADSL), le 802.11a culmine à 22 Mbit/s et le 802.11 g, enfin, flirte avec les 54 Mbit/s. Le Wi-Fi peut certes servir à surfer sur Internet, mais pas seulement. Il autorise l'organisation de réseaux pourvus ou pas d'Internet pour échanger des fichiers, des données,... Ce ne sont là que quelques exemples de ses usages possibles. Les avantages des réseaux sans fil ne sont plus à démontrer, surtout à une génération de plus en plus habituée à la mobilité. La multiplication des appareils (PDA, PC portables, terminaux et les téléphones portables) capables de communiquer entre eux en fait le support idéal

des réseaux modernes.

#### 1.1.1.2 Définition

Un réseau sans fil est un réseau qui permet de connecter différents nœuds sans l'aide d'une connexion physique, la communication est établie par des ondes électromagnétiques. C'est donc un réseau reliant au moins deux terminaux communiquant entre eux par voie Hertzienne, permettant un déplacement du terminal.

### 1.1.2 Classification des réseaux sans fil

Les réseaux sans fils peuvent être répartir suivants plusieurs critères de classification, mais le principal critère de classification est lié à leur zone de couverture. On distingue alors les WPAN, WLAN, WMAN, WWAN qui se différencient principalement grâce à leur portée maximale et les tehnologies utilisées pour communiquer.

- WPAN
  Ce sont des réseaux sans fil d'une faible portée (une dizaine de mètres) servant généralement à la connexion des périphériques (PDA, imprimante, téléphone portable, appareils domestiques,...), d'ordinateurs par les technologies telles que Blutooth, HomeRF et ZigBee.

- WLAN
  Il couvre l'équivalent d'un réseau local d'entreprise(100 m) et relie entre eux les équipements présents dans sa zone de couverture. Il utilise plus généralement la technologie Wi-Fi (IEEE 802.11) mais aussi l'HyperLan2 (HIgh Performance Radio LAN 2.0).

  - Le Wi-Fi (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance) offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres.
  - L'hiperLAN2 (HIgh Performance Radio LAN 2.0), norme européenne élaborée par l'ETSI (European Telecommunications Standards Institute), permet d'obtenir un débit théorique de 54 Mbps sur une zone d'une centaine de mètres dans la gamme de fréquence comprise entre 5 150 et 5 300 MHz.

- WMAN
  Encore appelé réseau métropolitain, il permet à un particulier ou une entreprise d'être relié à son opérateur (téléphonie fixe, Internet, télévision,...) grâce à la technologie WiMax.

- WWAN
  Plus connu sous le nom de «réseau cellulaire mobile» il est utilisé par les téléphones mobiles et est basé sur des technologies telles que : GSM, GPRS et UMTS.

  Le tableau suivant fait un récapitulatif des principaux critères de classification ci-dessus présentés.

| Cat. | Portée max | Débit | Usages | Normes |
|---|---|---|---|---|
| WPAN | Qqs m | 1 Mbit/s | Réseau particulier | IEEE 802.15 (Bluetooth), NFC, ETSI HyperPan |
| WLAN | 500 m | + de 50 Mbit/s | Réseaux internes, propres à un bâtiment (soit comme réseau d'entreprise, soit comme réseau domestique). | IEEE 802.11 (a,b,c,...) ETSI HyperLan |
| WMAN | 4 à 10 kilomètres | de 1 à 10 Mbit/s | Ville, Campus, ... Interconnecte plusieurs WLAN | IEEE 802.16 WiMax ETSI HyperMan |
| WWAN | Plusieurs centaines de kms | de 1 à 10 Mbit/s | Régional, National Interconnecte plusieurs villes | Basé sur des technologies cellulaires |

FIGURE 1.1 – Classification des réseaux sans fil
Source : slideplayer.fr

### 1.1.3 Avantages et Inconvénients des réseaux sans fil

#### 1.1.3.1 Avantages

Les réseaux sans fil permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres, l'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes comme c'est le cas avec les réseaux filaires.

#### 1.1.3.2 Inconvénients

En contrepartie, il se pose le problème des différentes réglementations relatives aux transmissions radio-électriques. De plus, les ondes hertziennes sont difficiles à confiner dans une surface géographique restreinte, C'est donc facile pour un pirate d'écouter le réseau si les informations circulent en clair. Il est donc nécessaire de mettre en place les dispositifs nécessaires de manière à assurer la confidentialité des données circulant sur les réseaux sans fil.

## 1.2 IEEE 802.11 (Wi-Fi)

### 1.2.1 Présentation

Conçue pour fournir un équivalent à Ethernet, la norme IEEE 802.11 est un standard international décrivant les caractéristiques d'un réseau local sans fil (WLAN). Le nom Wi-Fi correspond initialement au nom donnée à la certification délivrée par la Wi-Fi Alliance, l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11. Par abus de langage et aussi pour des raisons de marketing, le nom de la norme se confond aujourd'hui avec le nom de la certification. Ainsi, un réseau Wifi est en réalité un réseau répondant à la norme 802.11.
La norme 802.11 s'attache à définir les couches basses (les couches physiques et liaison de données) du modèle OSI pour une liaison sans fil utilisant des ondes électromagnétiques.
La couche physique définit la modulation des ondes radio-électriques et les caractéristiques de la signalisation pour la transmission de données, tandis que la couche liaison de données définit l'interface entre le bus de la machine et la couche physique, notamment une méthode d'accès proche de celle utilisée dans le standard Ethernet et les règles de communication entre les différentes stations.

### 1.2.2 Les différentes normes Wi-Fi

En réalité, la norme IEEE 802.11 est la norme initiale offrant des débits de 1 à 2 Mbps. Des modifications ont été apportées à celle-ci soit pour optimiser le débit, améliorer la sécurité ou encore pour une meilleure interopérabilité. L'image suivante retrace chronologiquement les différentes normes Wi-Fi depuis le 802.11.

| Norme WiFi | Date d'apparition | Fréquences | Largeur de canal | Débit maximum théorique | Portée | Nom |
|---|---|---|---|---|---|---|
| 802.11 | 1997 | 2,4Ghz | 20Mhz | 2Mbps | 20m | N/A |
| 802.11b | 1999 | 2,4Ghz | 20Mhz | 11Mbps | 35m | WiFi 1 |
| 802.11a | 1999 | 5Ghz | 20Mhz | 54Mbps | 35m | WiFi 2 |
| 802.11g | 2003 | 2,4Ghz | 20Mhz | 54Mbps | 38m | WiFi 3 |
| 802.11n | 2009 | 2,4Ghz ou 5Ghz | 20, 40Mhz | 72,2Mbps - 450Mbps | 70m | WiFi 4 |
| 802.11ac vague 1 | 2014 | 5Ghz | 20, 40, 80Mhz | 866,7Mbps | 35m | WiFi 5 |
| 802.11ac vague 2 | 2016 | 5Ghz | 20, 40, 80, 160Mhz | 1,73Gbps | 35m | WiFi 5 |
| 802.11ax | fin 2019 | 2,4Ghz ou 5Ghz | 20, 40, 80, 160Mhz | 2,4Gbps | N/A | WiFi 6 |

FIGURE 1.2 – Les différentes normes Wi-Fi
Source : selectra.info

### 1.2.3 Les équipements Wi-Fi

#### 1.2.3.1 Les adaptateurs sans fil ou carte d'accès

Aussi connu sous le nom de Network Interface Controler ou Network Interface Carte, il s'agit de cartes réseaux adaptées à la norme 802.11 et permettant à une station de se connecter à un réseau Wi-Fi. Il est disponible dans de nombreux formats : carte PCI, PCMCIA, adaptateur USB…



FIGURE 1.3 – Adaptateur USB
Source : www.rueducommerce.fr



FIGURE 1.4 – Carte wifi PCI
Source : www.amazon.fr

#### 1.2.3.2 Les points d'accès

Un point d'accès est donc un appareil qui crée un réseau local sans fil, ou WLAN, habituellement dans un bureau ou dans un grand bâtiment. Un point d'accès se connecte à un routeur filaire, commutateur ou hub par câble Ethernet et délivre un signal Wi-Fi à une zone dédiée.



FIGURE 1.5 – Un point d'accès sans fil
Source : slideplayer.fr

## 1.3 Architecture des réseaux Wi-Fi

Le standard 802.11 définit deux modes opératoires :

- Le mode Infrastructure dans lequel les clients sans fils sont connectés à un point d'accès. Il s'agit généralement du mode par défaut des cartes 802.11b.
- Le mode Ad-hoc dans lequel les clients sont connectés les uns aux autres sans aucun point d'accès.

### 1.3.1 Le mode infrastructure

En mode Infrastructure, chaque ordinateur station se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situés dans sa zone de couverture est appelé BSS et constitue une cellule. Chaque BSS est identifié par un BSSID, un identifiant de six octets. Dans le mode infrastructure, le BSSID correspond à l'adresse MAC du point d'accès.
La communication avec le point d'accès et une station désirant se connecter à celui-ci se fait à travers l'échange de deux types de paquets : les Beacons et les Probes.

- Le Beacon : Trame diffusée périodiquement par le point d'accès pour signaler sa présence. Elle contient des informations telles que le BSSID et les capacités de son réseau.
- le Probe : Requête diffusée par les stations afin de se connecter au point d'accès.

### 1.3.2 Le mode Ad-hoc

En mode Ad-hoc, les machines sans fil clientes se connectent les unes aux autres afin de constituer un réseau point à point, c'est-à-dire un réseau dans lequel chaque machine joue en même temps de rôle de client et le rôle de point d'accès. L'ensemble formé par les différentes stations est appelé IBSS. Un IBSS est ainsi un réseau sans fil constitué au minimum de deux stations et n'utilisant pas de point d'accès. L'IBSS constitue donc un réseau éphémère permettant à des personnes situées dans une même salle d'échanger des données.

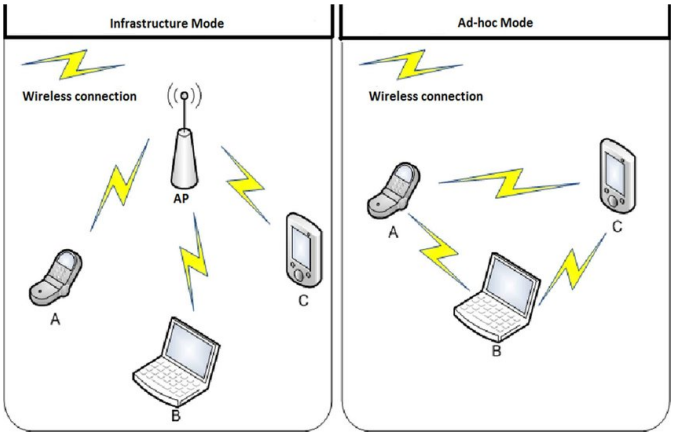



FIGURE 1.6 – Modes infrastructure et Ad-hoc
Source : www.researchgate.net

## 1.4 Sécurité des réseaux Wi-Fi

### 1.4.1 La sécurité informatique et ses principes fondamentaux

La sécurité informatique est l'ensemble des moyens mis en oeuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. L'objectif de la sécurité informatique est d'assurer que les ressources matérielles comme logicielles d'un parc informatique sont uniquement utilisées dans le cadre prévu et par des personnes autorisées. Voici les principaux objectifs de la sécurité informatique :

- La confidentialité des données : La confidentialité implique que seules les personnes ou entités autorisées peuvent accéder à des données spécifiques. Cela concerne notamment les informations personnelles, professionnelles ou sensibles.
- L'intégrité des données : L'intégrité des données implique que celles-ci restent correctes et cohérentes au fil du temps. Cela signifie qu'aucune altération ou corruption non autorisée ne doit se produire.

- La disponibilité des données : La disponibilité signifie que les ressources informatiques, y compris les données et les applications, sont opérationnelles et accessibles à tout moment, sans interruption non planifiée.

### 1.4.2 Les attaques contre les réseaux sans fil

Les attaques perpétrées contre les réseaux sans fil ont généralement pour but de d'accéder frauduleusement au réseau ou de rendre celui-ci indisponible portant ainsi atteinte à deux des principes fondamentaux de la sécurité informatique : la confidentialité et la disponibilité.

#### 1.4.2.1 Denial of Service

Le déni de service consiste à rendre inopérant un système afin d'en empêcher l'accès à des utilisateurs réguliers. Cela consiste, par exemple, à saturer le point d'accès en multipliant artificiellement le nombre de demandes. Le point d'accès considère alors que de nombreuses machines veulent se connecter. Par conséquent, la surconsommation d'énergie due à l'obligation de répondre aux sollicitations de l'attaquant provoque un affaiblissement rapide des batteries, c'est ce que l'on appelle un déni de service sur batterie. Il s'agit d'une attaque sur la disponibilité.

#### 1.4.2.2 Le spooffing

Le spooffing consiste à usurper soit l'adresse IP, soit l'adresse MAC d'une autre machine. En modifiant l'adresse IP source dans l'entête du paquet, le récepteur croira avoir reçu un paquet de cette machine. Si le serveur considérait cette machine comme une machine de confiance, beaucoup de données sensibles pourraient être consultées, modifiées, voire supprimées.

#### 1.4.2.3 Le Men-In-The-Middle en milieu Wi-Fi

Cette attaque consiste, pour un réseau Wi-Fi, à disposer un point d'accès étranger à proximité du point d'accès légitime. Les stations désirant se connecter au réseau livreront au leurre leurs informations de connexion. Ces informations pourront être utilisées par une station pirate pour se connecter au réseau.

À ces différentes menaces, plusieurs solutions ont été développées afin de lutter efficacement contre les violations d'accès aux réseaux sans fils.

### 1.4.3 Solutions de sécurisation des réseaux sans fil

#### 1.4.3.1 Le WEP

Créé en septembre 1999, il est l'une des premières solutions de restriction d'accès aux réseaux sans fil. Il permet de crypter la connexion par l'intermédiaire d'un mot de passe, aussi appelé "clé WEP". Cependant, ce protocole a rapidement montré ses limites, laissant un niveau encore trop élevé de vulnérabilité sur les réseaux Wi-Fi. Il fut progressivement remplacé par le WPA.

#### 1.4.3.2 le WPA

Le WPA est une amélioration de l'algorithme WEP et de l'authentification des réseaux 802.11. Développé par l'IEEE pour combler les faiblesses du WEP, le WPA offre une sécurité nettement supérieure

par rapport au WEP grâce a :

- des techniques de cryptage plus aléatoires : dans WPA, contrairement au WEP, le caractère aléatoire du cryptage est nettement renforcé, ce qui a pour effet de nettement compliquer la tache du pirate ;
- de grande facilité d'utilisation : avec le WPA, l'utilisateur n'aura pas de problème concernant la représentation de la clé qui doit être une fois en hexadécimal, une autre en ASCII. Ici , il ne faut utiliser qu'un simple mot de passe.

Cette technologie à subit des améliorations et est à sa troisième version (WPA2,WPA3).

#### 1.4.3.3 L'architecture 802.1x

Le protocole 802.1x est une solution standard de sécurisation réseaux mise au point par l'IEEE en 2001. 802.1x permet d'authentifier un utilisateur souhaitant accéder à un réseau grâce à un serveur central d'authentification (serveur AAA).

## Conclusion

Les réseaux sans fil permettent de relier des ordinateurs et d'autres appareils informatiques sans avoir à installer de câblage, ce qui représente plus de confort et fait économiser de l'argent au niveau des infrastructures. En revanche, ce genre de réseaux a généralement moins de sécurité, car sans protection efficace, l'introduction d'intrus est très probable.

Chapitre 2

# Analyse choix technique et Conception

## Introduction

Face aux différentes menaces liées à la sécurité des réseaux Wi-Fi, de diverses solutions ont été développées. Dans ce chapitre, il s'agira de dire l'outil sur lequel notre choix s'est porte, de décrire son fonctionnement ainsi que les facteurs qui ont motivé son choix.

## 2.1 Généralités sur la solution choisie

### 2.1.1 Choix de l'architecture de sécurisation

Après analyse et comparaison des solutions existantes, nous avons opté pour l'architecture 802.1x. En effet, le protocole 802.1x est une solution standard de sécurisation de réseaux mise au point par l'IEEE en 2001.IL permet d'authentifier un utilisateur souhaitant accéder à un réseau (câblé ou Wi-Fi) grâce à un serveur d'authentification sur lequel sont centralisées les données de d'identification et d'authentification des utilisateurs. Ceci a pour avantage de savoir exactement qui se connecte et quand ; contrairement aux WEP et WPAs où il faut juste connaître le mot de passe.

### 2.1.2 Architecture du 802.1x

Comme on peut le voir dans au niveau de la figure 2.1, le protocole 802.1x est composé de trois entités qui interagissent pour le processus d'authentification :

- Le système a authentifié ou Supplicant
  Ce sont les postes clients (ordinateurs, tablettes, téléphones portables, imprimante,...) qui disposent du protocole 802.1X et peuvent ainsi assurer la transmission des informations vers le NAC (Network Access Server : équipement permettant de recueillir les informations clients et de les retransmettre au serveur d'authentification.

- Le client Radius
  C'est un équipement réseau (commutateur, routeur, borne wifi,...) qui agit comme une barrière de sécurité entre le supplicant et le réseau protégé. Il sert de relais entre le Supplicant et le serveur d'authentification et gère le PAE (Port Access Entity) qui permet au Supplicant d'accéder

ou non au réseau ;

- Le serveur d'authentification (Radius)

  RADIUS (Remote Authentication Dial-In User Service) est un protocole client-serveur permettant de centraliser des demandes d'authentification relayées par des équipements de réseau, comme des commutateurs ou bornes Wifi, considérés alors comme ses clients. Radius centralise les informations concernant les utilisateurs afin de les authentifier sur un réseau. Cette authentification se basera soit sur un couple identifiant/mot de passe, soit sur des certificats. Il permet aussi l'accounting (comptabilisation) afin de journaliser les accès.

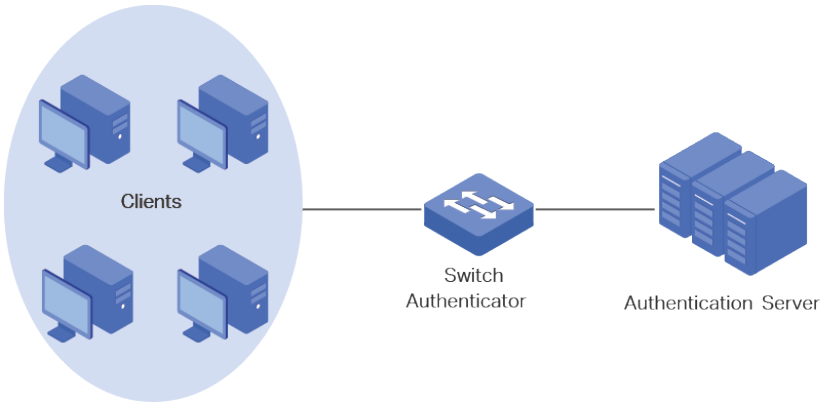



FIGURE 2.1 – Les 3 entités qui interagissent dans le protocole 802.1x
Source : www.tp-link.com

### 2.1.3 Les méthodes d'authentification du 802.1x

Le protocole 802.1X définit l'utilisation d'EAP (Extensible Authentication Protocol), mécanisme décrivant la méthode utilisée pour réaliser l'authentification. On distingue deux types de trafic EAP (figure 2-2) :

- entre le système à authentifier et le point d'accès (support : 802.11a, b, g ou 802.3) : EAP over LAN (EAPOL)
- entre le point d'accès et le serveur d'authentification (de type RADIUS) : EAP over Radius.

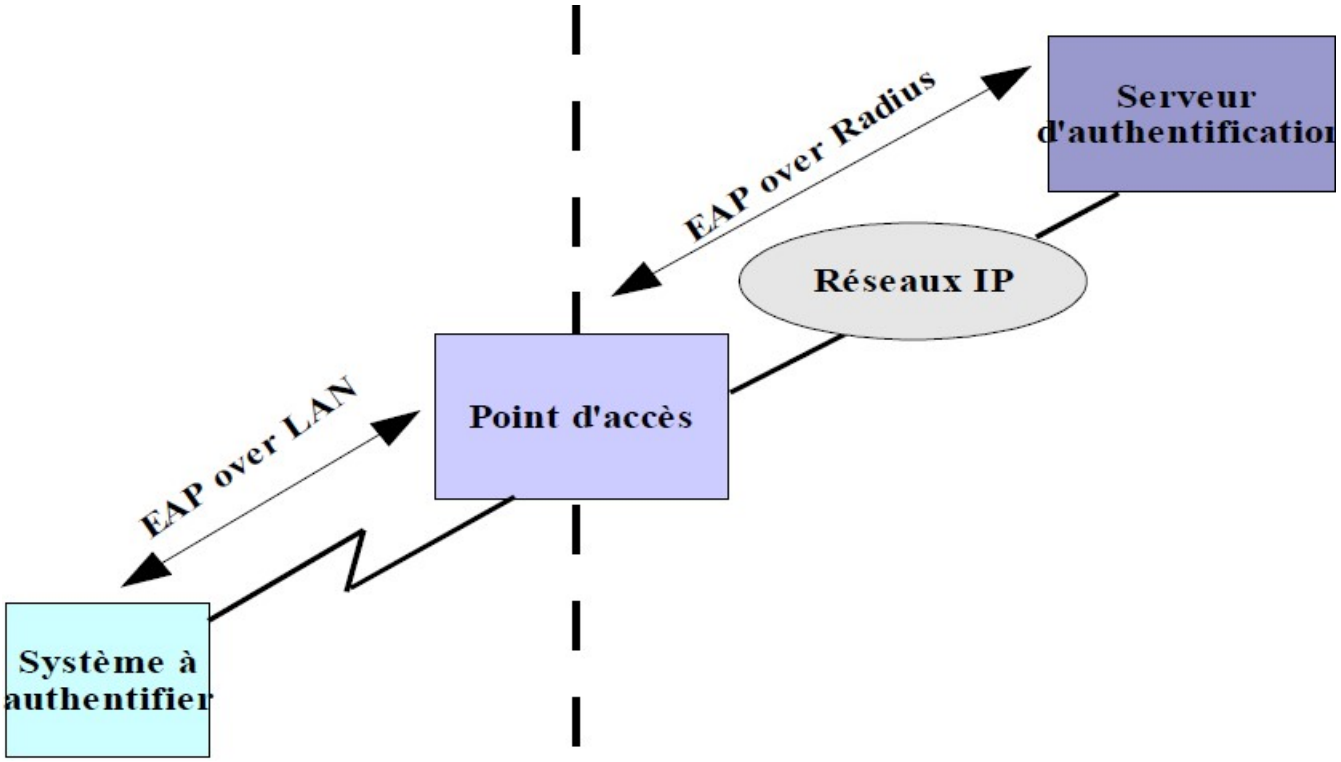



FIGURE 2.2 – Les trafics EAP
Source : www.memoireonline.com

Les messages EAP se décomposent en quatre classes :

- requêtes (du serveur vers le client) ;
- réponses (du client vers le serveur) ;
- succès ;
- erreur ou échec.

Comme le montre la figure 2-3, la première étape est l'association physique avec le point d'accès (équivalent au branchement d'un câble reliant le port d'un commutateur à l'équipement à authentifier) qui doit être réalisée préalablement à la phase d'authentification 802.1X. Le processus d'authentification est ensuite initié par l'envoi d'une requête provenant du point d'accès vers le client (EAPOL). Le client répond à la requête en y joignant un premier identifiant (nom de la machine, login, etc). Cette réponse est retransmise au serveur Radius (EAP over Radius). À partir de ce moment, les échanges dépendent de la méthode d'authentification choisie (EAP-TLS, LEAP, etc.). Au terme de ces échanges, le client est soit authentifié, auquel cas le point d'accès autorise le trafic du client sur le réseau, soit non authentifié, et l'accès au réseau reste alors interdit. À la fin d'un scénario d'authentification réussi, Supplicant et serveur d'authentification calculent un secret partagé, baptisé Unicast Key. Le serveur d'authentification transmet cet élément, grâce au protocole RADIUS, au point d'accès. Ce dernier choisit alors une clé WEP, et la pousse vers le Supplicant, de manière sécurisée, dans une trame EAPoL-Key, chiffrée et signée à l'aide de la clé Unicast.

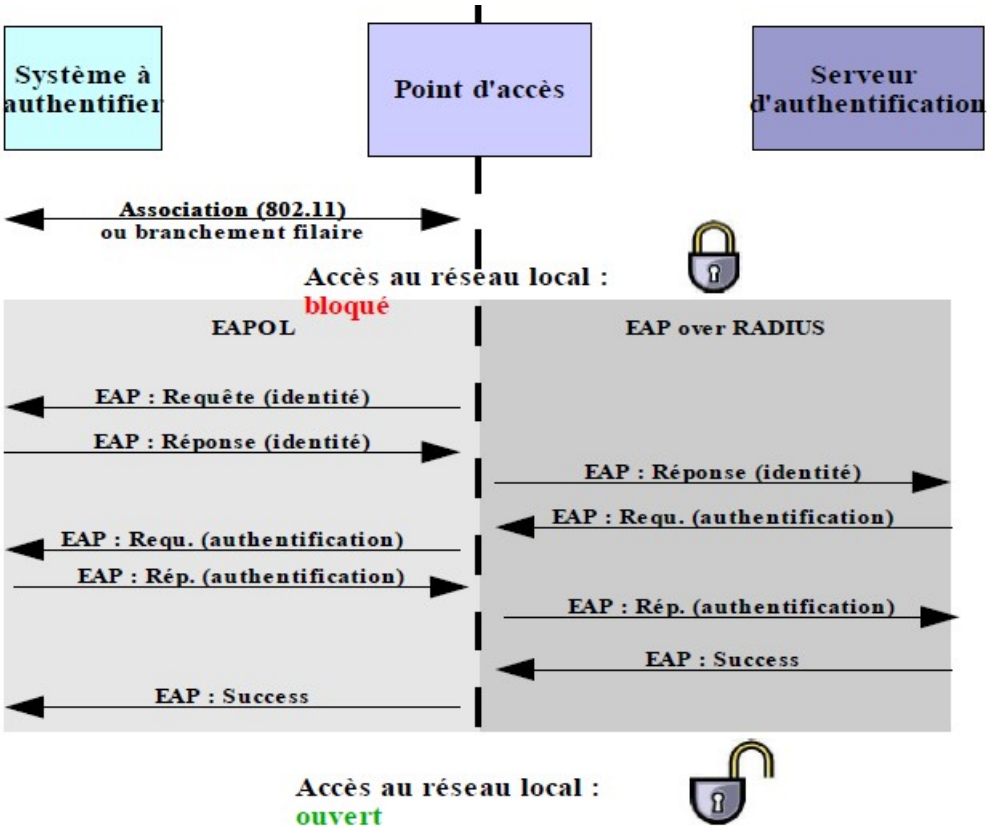



FIGURE 2.3 – Les étapes d'une connexion EAP
Source : www.memoireonline

### 2.1.4 Les méthodes associées a EAP

Le protocole 802.1X ne propose pas une seule méthode d'authentification, mais un canevas sur lequel sont basés plusieurs types d'authentification. Ainsi, une méthode d'authentification EAP utilise différents éléments pour identifier un client :

- login / mot de passe ;
- certificat électronique ;
- biométrie ;
- puce (SIM).

Certaines méthodes combinent plusieurs critères (certificats et login/mot de passe, etc.). En plus de l'authentification, EAP gère la distribution dynamique des clés de chiffrement (WEP). Encapsulé dans le protocole RADIUS, EAP est utilisé pour transporter les informations d'authentification entre deux équipements. Les différentes méthodes sont entre autres : PAP, CHAP, MD-5, TLS,...(figure 2.4).

#### 2.1.4.1 EAP-PAP

C'est le premier protocole qui a été utilisé. Avec PAP les mots de passe circulaient en clair, ce qui ne rend pas cette méthode très fiable.

#### 2.1.4.2 EAP-MD5

Le type EAP-MD5 (Message Digest) qui est un type d'authentification EAP qui fournit une prise en charge EAP au niveau de base. Le protocole EAP-MD5 n'est généralement pas recommandé dans les implémentations de réseau Wi-Fi, car il est possible que le mot de passe de l'utilisateur soit dérivé. Il ne fournit qu'une authentification à sens unique : il n'existe pas d'authentification mutuelle du client Wi-Fi et du réseau. Et, ce qui est très important, il ne fournit pas de moyen d'obtenir des clés WEP dynamiques par session.

#### 2.1.4.3 EAP-TLS

L'authentification EAP-TLS qui prévoit une authentification mutuelle et basée sur un certificat du client et du réseau. Il s'appuie sur des certificats côté client et côté serveur pour effectuer l'authentification et peut-être utilisé pour générer dynamiquement des clés WEP basées sur l'utilisateur et les sessions afin de sécuriser les communications ultérieures entre le client WLAN et le point d'accès. Un inconvénient de l'EAP-TLS est que les certificats doivent être gérés à la fois du côté du client et du serveur. Pour une installation de réseau WLAN de grande taille, cela peut être une tâche très lourde.

#### 2.1.4.4 EAP-TTLS

Le protocole EAP-TTLS qui a été développé par Funk Software et Certicom, dans le cadre d'une extension de l'EAP-TLS. Cette méthode de sécurité prévoit une authentification mutuelle du client et du réseau basée sur un certificat via un canal crypté (ou tunnel), ainsi qu'un moyen d'utiliser des clés WEP dynamiques, par utilisateur et par session. Contrairement à l'EAP-TLS, le type EAP-TTLS ne nécessite que des certificats côté serveur.

#### 2.1.4.5 EAP-FAST

EAP-FAST qui a été développé par Cisco. Au lieu d'utiliser un certificat pour obtenir une authentification mutuelle, EAP-FAST authentifie au moyen d'un PAC (Protected Access Credential) qui peut être géré dynamiquement par le serveur d'authentification. Le PAC peut être approvisionné (distribué une fois) au client manuellement ou automatiquement. L'approvisionnement manuel est la livraison au client par disque ou une méthode de distribution réseau sécurisée. L'approvisionnement automatique est une distribution en bande, sans fil.

#### 2.1.4.6 EAP-SIM

La méthode d'authentification du protocole extensible pour l'identité d'abonné GSM (EAP-SIM) est un mécanisme d'authentification et de distribution des clés de session. Elle utilise le module SIM du système de communication GSM. EAP-SIM utilise une clé WEP dynamique de session, dérivée de la carte client et du serveur RADIUS, pour chiffrer les données. EAP-SIM requiert la saisie d'un code de vérification utilisateur ou PIN, pour communiquer avec la carte SIM.

#### 2.1.4.7 EAP-AKA

La méthode EAP-AKA qui est un mécanisme EAP pour l'authentification et la distribution des clés de session, utilisant le module d'identification d'abonné USIM de Universal Mobile Telecommunications System (UMTS). La carte USIM est une carte à puce spéciale utilisée avec les réseaux cellulaires pour valider un utilisateur donné avec le réseau.

#### 2.1.4.8 EAP- LEAP

Le protocole LEAP (Lightweight Extensible Authentication Protocol) qui est un type d'authentification EAP utilisé principalement dans les WLAN Cisco Aironet. Il chiffre les transmissions de données à l'aide de clés WEP générées dynamiquement et prend en charge l'authentification mutuelle. Cisco a accordé des licences pour l'adéquation de la technologie LEAP à une variété d'autres fabricants dans le cadre de son programme Cisco Compatible Extensions.

#### 2.1.4.9 EAP-PEAP

Le protocole PEAP qui fournit une méthode permettant de transporter en toute sécurité les données d'authentification, y compris les protocoles anciens basés sur mot de passe, sur les réseaux Wi-Fi 802.11. Le protocole PEAP y parvient en utilisant un tunnel entre les clients PEAP et un serveur d'authentification. À l'instar de la norme concurrente Tunneled Transport Layer Security (TTLS), le protocole PEAP authentifie les clients de réseaux Wi-Fi en utilisant uniquement des certificats côté serveur, simplifiant ainsi la mise en œuvre et l'administration d'un réseau Wi-Fi sécurisé. Microsoft, Cisco et RSA Security ont développé le protocole PEAP.

| **Types EAP 802.1X**<br>**Fonctionnalités /avantages** | **MD5**<br>---<br>**Digest des messages 5** | **TLS**<br>---<br>**Sécurité au niveau du transport** | **TTLS**<br>---<br>**Sécurité au niveau du transport tunnelé** | **PEAP**<br>---<br>**Sécurité au niveau du transport protégé** | **RAPIDE**<br>---<br>**Authentification flexible via tunnel sécurisé** | **SAUT**<br>---<br>**Protocole d'authentification extensible léger** |
|---|---|---|---|---|---|---|
| Certificat côté client requis | non | Oui | non | non | non (PAC) | non |
| Certificat côté serveur requis | non | Oui | Oui | Oui | non (PAC) | non |
| Gestion des clés WEP | non | Oui | Oui | Oui | Oui | Oui |
| Détection des points d'accès non autorisés | non | non | non | non | Oui | Oui |
| Fournisseur | MS | MS | Funk | MS | Cisco | Cisco |
| Attributs d'authentification | À sens unique | Mutuelle | Mutuelle | Mutuelle | Mutuelle | Mutuelle |
| Difficultés de déploiement | Facile | Difficile (en raison du déploiement du certificat client) | Modérée | Modérée | Modérée | Modérée |
| Sécurité Wi-Fi | Pauvres | Très élevé | Haute | Haute | Haute | Élevée lorsque des mots de passe forts sont utilisés. |

FIGURE 2.4 – Etude comparative de quelques types du protocole d'authentification EAP
Source : www.intel.ca

## 2.2 Analyse du choix du type d'authentification EAP-TLS

EAP-TLS est une méthode s'appuie sur les certificats électroniques. L'utilisation des certificats apporte plus de sûreté, car ils sont plus sûrs que les mots de passe. Cela permet aussi d'éviter le partage de bouche à oreille des mots de passe. Qu'est-ce qu'un certificat électronique en réalité ? et comment ça fonctionne ?

### 2.2.1 Le certificat numérique ou électronique

Les certificats numériques permettent l'identification unique d'une entité ; il s'agit de cartes d'identité électronique émises par des parties dignes de confiance. Les certificats numériques permettent à un utilisateur de vérifier l'identité de la personne à laquelle le certificat est délivré ainsi que l'identité de l'émetteur du certificat. L'autorité de certification fait foi de tiers de confiance et atteste du lien entre l'identité physique et l'entité numérique.

### 2.2.2 Authentification par les certificats

Les certificats numériques sont le vecteur utilisé par SSL pour la cryptographie de clé publique. La cryptographie de clé publique utilise deux clés cryptographiques différentes : une clé privée et une clé publique. Les paires de clés publiques-privées sont simplement de longues chaînes de données qui jouent le rôle de clés dans le schéma de chiffrement d'un utilisateur. L'utilisateur conserve la clé privée dans un endroit sécurisé (par exemple, le disque dur d'un ordinateur) et fournit la clé publique à quiconque avec qui il souhaite communiquer. La clé privée permet de doter d'une signature numérique toutes les communications sécurisées émanant de l'utilisateur ; la clé publique est utilisée par le destinataire pour vérifier la signature de l'expéditeur. La cryptographie de clé publique s'appuie sur la confiance ; en effet, le destinataire d'une clé publique doit être prêt à croire que la clé appartient réellement à l'expéditeur et non à un imposteur. Les certificats numériques offrent cette confiance.
Le rôle d'un certificat numérique est double : il établit l'identité du propriétaire d'une part, et il rend la clé publique du propriétaire disponible d'autre part. Un certificat numérique est émis par une autorité digne de confiance (une autorité de certification (AC)) et uniquement pour une durée déterminée. Une fois sa date d'expiration passée, le certificat numérique doit être remplacé.

Mis à part les avantages qu'apportent les certificats, il serait important de remarque que dans EAP-TLS l'authentification par les certificats se fait à la fois du côté du serveur RADIUS que du client désirant se connecter au réseau, ce qui permet d'éviter par exemple les attaques de types Men-in-the-Middle : on parle d'authentification mutuelle.

### 2.2.3 Déroulement de EAP-TLS

EAP s'appuie sur TLS pour proposer une authentification sécurisée. Cette méthode s'appuie sur les certificats électroniques. Ainsi, chaque partie (serveur et client) doit posséder un certificat pour prouver son identité. Les différentes étapes d'une authentification EAP-TLS sont représentées dans la figure ci-dessous suivie d'une description.

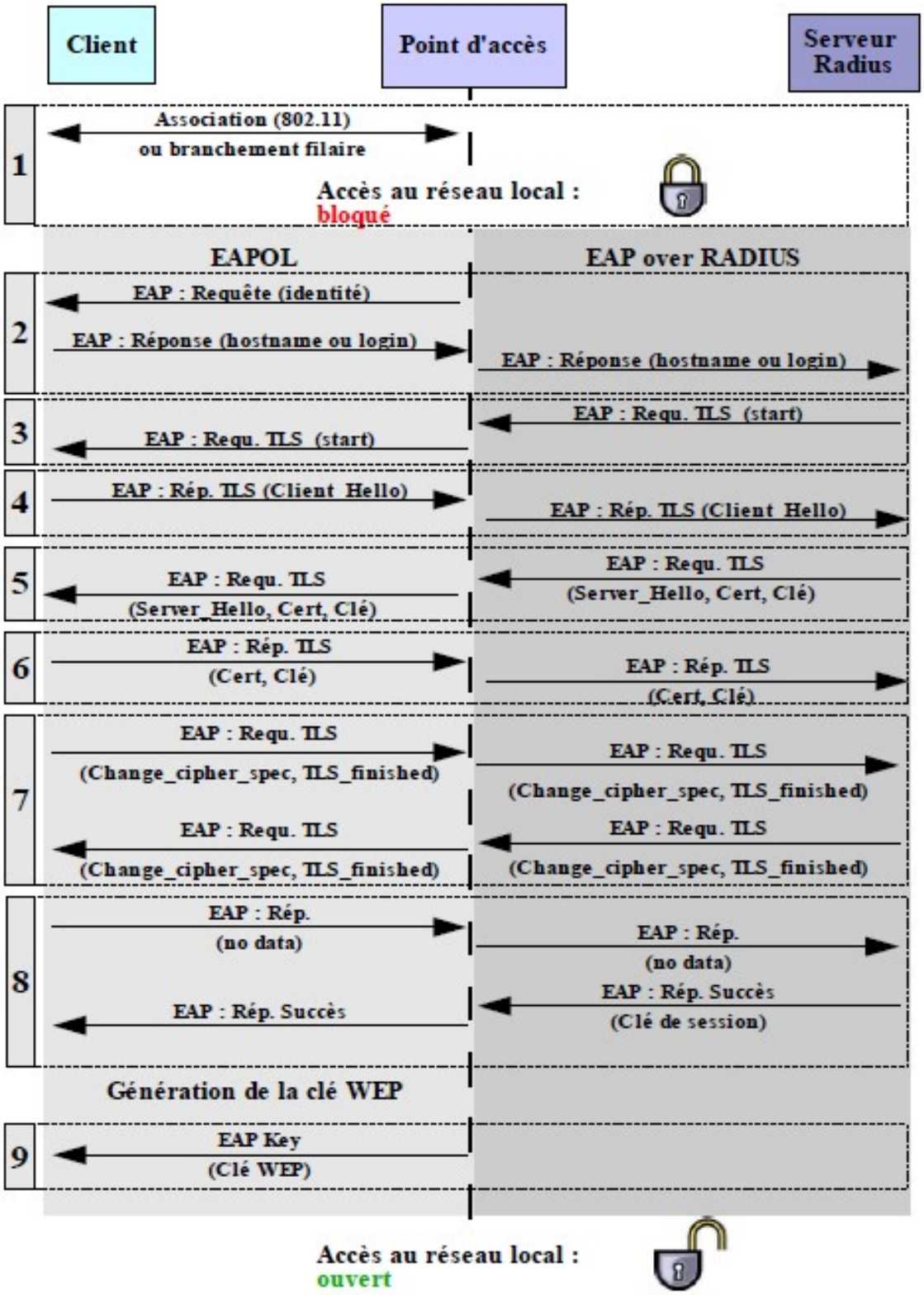



FIGURE 2.5 – Etapes d'une authentification EAP-TLS
Source : www.memoireonline.com

1) Le client s'associe physiquement au point d'accès.
2) Le point d'accès envoie une requête d'authentification au client. Le client répond avec son identifiant (nom de machine ou login), ce message est relayé par le point d'accès vers le serveur Radius.
3) Le serveur Radius initie le processus d'authentification TLS par le message TLS start.
4) Le client répond avec un message clienthello, qui contient :

- des spécifications de chiffrement vides en attendant qu'elles soient négociées entre le client et le serveur ;
- la version TLS du client ;
- un nombre aléatoire (défi ou challenge) ;
- un identifiant de session ;
- les types d'algorithmes de chiffrement supportés par le client.

5) Le serveur renvoie une requête contenant un message serverhello suivi :

- de son certificat (x509) et de sa clé publique ;
- de la demande du certificat du client ;
- d'un nombre aléatoire (défi ou challenge) ;
- d'un identifiant de session (en fonction de celui proposé par le client).

Le serveur choisit un algorithme de chiffrement parmi ceux qui lui ont été proposés par le client.
6) Le client vérifie le certificat du serveur et répond avec son propre certificat et sa clé publique.
7) Le serveur et le client, chacun de son côté, définissent une clé de chiffrement principale utilisée pour la session. Cette clé est dérivée des valeurs aléatoires que se sont échangées le client et le serveur. Les messages changecipherspec indiquent la prise en compte du changement de clé. Le message TLSfinished termine la phase d'authentification TLS (TLS handshake), dans le cas d'EAP-TLS la clé de session ne sert pas à chiffrer les échanges suivants.
8) Si le client a pu vérifier l'identité du serveur (avec le certificat et la clé publique), il renvoie une réponse EAP sans donnée. Le serveur retourne une réponse EAP success.
9) La clé de session générée en (8) est réutilisée par le point d'accès pour créer une clé WEP qui est transmise au client, dans le cas où il s'agit d'une station Wifi. La clé de session est valide jusqu'à ce que le client se déconnecte ou que son authentification expire, auquel cas, il doit s'identifier à nouveau. Le tunnel TLS créé lors de la création de la clé de session n'est donc pas exploité. Seul le TLS Handshake est utilisé, il permet l'authentification mutuelle des deux parties.
EAP-TLS est une méthode d'authentification performante. Seuls les problèmes liés aux IGC peuvent dissuader de l'utilisation de cette méthode. EAP-TLS est une méthode d'authentification performante. Seuls les problèmes liés aux IGC (Infrastructure de Gestion de Clé) peuvent dissuader de l'utilisation de cette méthode.

## Conclusion

Dans ce chapitre, il était question de choisir une solution et d'analyser son choix. On a choisi EAP-TLS car il assure l'authentification non seulement d'une mais des deux entités à base de certificats qui sont plus sûrs que les mots de passe.

Chapitre 3

# Résultat obtenu et Perspectives

## Introduction

Dans ce chapitre, il s'agira d'installer un serveur FreeRadius sur une machine Ubuntu, d'y installer EAP-TLS puis d'effectuer des tests après avoir mis en place un environnement de test.

## 3.1 Installation de FreeRadius

Dans un premier temps, il faudra procéder à l'implémentation de freeRadius dont l'installation complète nécessite le téléchargement d'un package directement sur leur plateforme. En effet, la version par défaut contenu dans "aptlist" n'installe pas tous les fichiers nécessaires. Le contenu du package est le suivant :

```
root@ksbleck-HP-Laptop-15-dw3xxx:/usr/local/freeradius/freeradius-server-3.2.0/f
reeradius-server-3.2.0# ls
acinclude.m4   config.sub    debian       main.mk      missing      site.yml
aclocal.m4     configure     doc          Makefile     raddb        src
build          configure.ac  INSTALL.rst  Make.inc     README.rst   suse
config.guess   CONTRIBUTING  install-sh    Make.inc.in  redhat       VERSION
config.log     COPYRIGHT     LICENSE      man          scripts
config.status  CREDITS       m4           mibs         share
```

FIGURE 3.1 – Contenu du package nécessaire à l'installation de FreeRadius

On passe maintenant à l'exécution du fichier install.sh de notre package d'installation de FreeRadius ce qui lancera l'installation de FreeRadius.

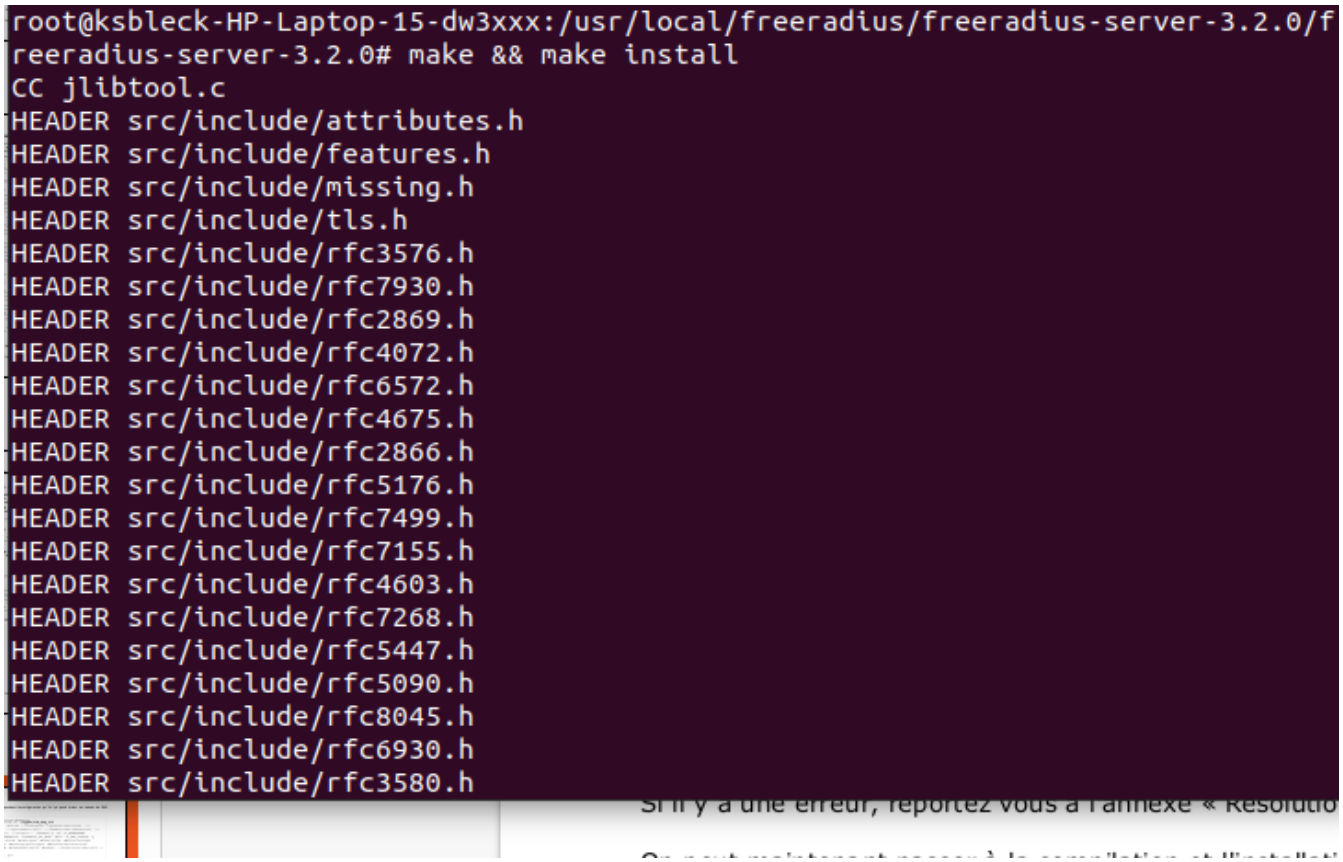



FIGURE 3.2 – Exécution du fichier install de notre package d'installation de FreeRadius

Après l'installation de FreeRadius, on peut voir ci-dessous différentes informations la concernant.

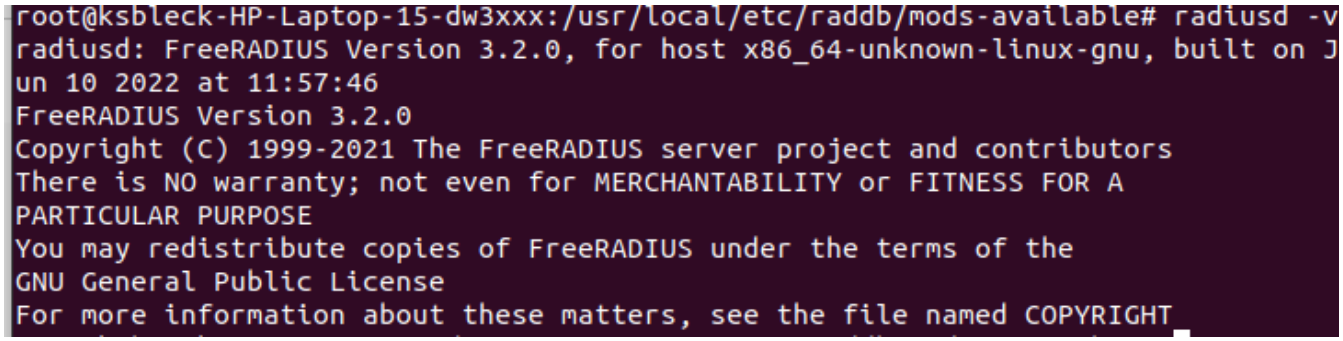
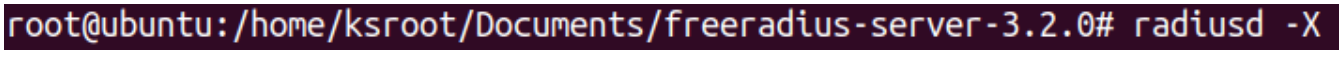



FIGURE 3.3 – Version de freeradius installée

Pour être sûr que FreeRadius marche correctement, on lance le daemon avec le debug.



```
Listening on auth address 127.0.0.1 port 18120 bound to server inner-tu
nnel
Listening on auth address * port 1812 bound to server default
Listening on acct address * port 1813 bound to server default
Listening on auth address :: port 1812 bound to server default
Listening on acct address :: port 1813 bound to server default
Listening on proxy address * port 55768
Listening on proxy address :: port 50038
Ready to process requests
```

FIGURE 3.4 – lancement de FreeRadius

Le daemon s'est bien lancé.

## 3.2 Implémentation de EAP-TLS

### 3.2.1 Génération des différents certificats

- Certificat racine

  Nous allons passer à la génération de notre certificat racine qui sera utilisé pour la signature du certificat client et du certificat serveur sur notre serveur d'authentification FreeRadius. Pour cela, on éditera le fichier "ca.cnf". Il s'agira de mentionner entre autres le pays où sera généré le certificat, la ville, le nom du certificat... Il en sera de même lors de la génération respective des certificats serveurs et client.

```
 GNU nano 4.8                          ca.cnf                          Modifié
input_password          = whatever
output_password         = whatever
x509_extensions         = v3_ca

[certificate_authority]
countryName             = BE
stateOrProvinceName     = Benin
localityName            = Cotonou
organizationName        = Example Inc.
emailAddress            = admin@example.org
commonName              = "Example Certificate Authority"

[v3_ca]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always,issuer:always
basicConstraints        = critical,CA:true
crlDistributionPoints   = URI:http://www.example.org/example_ca.crl
```

FIGURE 3.5 – Début de la génération du certificat racine

Nous allons ensuite passer à la génération proprement dite du certificat racine ci-dessous avec la commande "make" qui exécute la commande OpenSSL permettant de générer le certificat.

```
root@ksbleck-HP-Laptop-15-dw3xxx:/usr/local/freeradius/freeradius-server-3.2.0/
freeradius-server-3.2.0/raddb/certs# make ca.pem
openssl req -new -x509 -keyout ca.key -out ca.pem \
        -days '60' -config ./ca.cnf \
        -passin pass:'whatever' -passout pass:'whatever'
Generating a RSA private key
.....................................................+++++
.............................+++++
writing new private key to 'ca.key'
-----
chmod g+r ca.key
```

FIGURE 3.6 – Génération du certificat racine

```
root@ksbleck-HP-Laptop-15-dw3xxx:/usr/local/freeradius/freeradius-server-3.2.0/
freeradius-server-3.2.0/raddb/certs# make ca.der
openssl x509 -inform PEM -outform DER -in ca.pem -out ca.der
```

FIGURE 3.7 – Convertion du certificat racine dans un fichier binaire

- Certificat serveur

  La génération de ce certificat passe également par un renseignement d'informations liées au serveur. Cette modification se fera dans "server.cnf".

```
  GNU nano 4.8                          server.cnf
default_bits            = 2048
input_password          = whatever
output_password         = whatever
#req_extensions         = v3_req

[server]
countryName             = BE
stateOrProvinceName     = Benin
localityName            = Cotonou
organizationName        = Example Inc.
emailAddress            = admin@example.org
commonName              = "Example Server Certificate"

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
```

FIGURE 3.8 – Début de génération du certificat serveur

Nous allons passer à la signature du certificat serveur par le certificat racine précédemment généré.

```
freeradius-server-3.2.0/raddb/certs# make server.pem
openssl req -new  -out server.csr -keyout server.key -config ./server.cnf
Generating a RSA private key
.....................................................................
..........+++++
.....................................+++++
writing new private key to 'server.key'
-----
chmod g+r server.key
openssl ca -batch -keyfile ca.key -cert ca.pem -in server.csr  -key 'whatever'
-out server.crt -extensions xpserver_ext -extfile xpextensions -config ./server
.cnf
Using configuration from ./server.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jun  7 09:13:58 2022 GMT
            Not After : Aug  6 09:13:58 2022 GMT
        Subject:
            countryName               = BE
```

FIGURE 3.9 – Génération du certificat serveur

```
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
            X509v3 CRL Distribution Points:

                Full Name:
                  URI:http://www.example.com/example_ca.crl

            X509v3 Certificate Policies:
                Policy: 1.3.6.1.4.1.40808.1.3.2

Certificate is to be certified until Aug  6 09:13:58 2022 GMT (60 days)

Write out database with 1 new entries
Data Base Updated
openssl pkcs12 -export -in server.crt -inkey server.key -out server.p12  -passi
n pass:'whatever' -passout pass:'whatever'
chmod g+r server.p12
openssl pkcs12 -in server.p12 -out server.pem -passin pass:'whatever' -passout
pass:'whatever'
chmod g+r server.pem
```

FIGURE 3.10 – Confirmation

- certificat client

  Ici, c'est le fichier "client.cnf" qui sera édité.

```
  GNU nano 4.8                          client.cnf
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ req ]
prompt                  = no
distinguished_name      = client
default_bits            = 2048
input_password          = whatever
output_password         = whatever

[client]
countryName             = BE
stateOrProvinceName     = Benin
localityName            = Cotonou
organizationName        = Example Inc.
emailAddress            = user@example.org
commonName              = user@example.org
```

FIGURE 3.11 – Début de génération du certificat client

On passe à la signature du certificat client par le certificat racine précédemment généré.

```
freeradius-server-3.2.0/raddb/certs# make client.pem
openssl ca -batch -keyfile ca.key -cert ca.pem -in client.csr  -key 'whatever'
-out client.crt -extensions xpclient_ext -extfile xpextensions -config ./client
.cnf
Using configuration from ./client.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 2 (0x2)
        Validity
            Not Before: Jun  7 09:17:23 2022 GMT
            Not After : Aug  6 09:17:23 2022 GMT
        Subject:
            countryName               = BE
            stateOrProvinceName       = Benin
            organizationName          = Example Inc.
            commonName                = user@example.org
            emailAddress              = user@example.org
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
            X509v3 CRL Distribution Points:
```

FIGURE 3.12 – Génération du certificat client

```
                Full Name:
                  URI:http://www.example.com/example_ca.crl

Certificate is to be certified until Aug  6 09:17:23 2022 GMT (60 days)

Write out database with 1 new entries
Data Base Updated
openssl pkcs12 -export -in client.crt -inkey client.key -out client.p12  -passi
n pass:'whatever' -passout pass:'whatever'
chmod g+r client.p12
cp client.p12 'user@example.org'.p12
openssl pkcs12 -in client.p12 -out client.pem -passin pass:'whatever' -passout
pass:'whatever'
chmod g+r client.pem
cp client.pem 'user@example.org'.pem
```

FIGURE 3.13 – Confirmation

### 3.2.2 Génération des paramètres de l'algorithme de Diffie-Hellman

L'algorithme Diffie-Hellman est un algorithme d'échange de clés qui est utilisé pour chiffrer la communication entre les trois entités du système.

```
root@ksbleck-HP-Laptop-15-dw3xxx:/usr/local/freeradius/freeradius-server-3.2.0/f
reeradius-server-3.2.0/raddb/certs# make dh
openssl dhparam -out dh -2 2048
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
.........+.+.....................................................................
```

FIGURE 3.14 – Génération des paramètres de l'algorithme de Diffie-Hellman

### 3.2.3 Modification du fichier de configuration par défaut du protocole EAP

```
 GNU nano 4.8                          eap                          Modifié
        #  type they will be using, so it MUST be set here.
        #
        #  For now, only one default EAP type may be used at a time.
        #
        #  If the EAP-Type attribute is set by another module,
        #  then that EAP type takes precedence over the
        #  default type configured here.
        #
        default_eap_type = tls

        #  A list is maintained to correlate EAP-Response
        #  packets with EAP-Request packets.  After a
        #  configurable length of time, entries in the list
        #  expire, and are deleted.
        #
        timer_expire = 60
```

FIGURE 3.15 – Passage à eap-tls

### 3.2.4 Implémentation des chemins relatifs aux certificats dans le fichier de EAP

```
GNU nano 4.8                          eap                          M
       #  ANYONE who has a certificate signed by them can
       #  authenticate via EAP-TLS!  This is likely not what you want.
       #
       tls-config tls-common {
              private_key_password = whatever
              private_key_file = /usr/local/etc/raddb/certs/server.pem
```

FIGURE 3.16 – Implémentation du path du certificat serveur

```
                              eap                              M
#  There are many versions of software, each of which
#  behave slightly differently.  It is impossible to
#  give advice which will work everywhere.  Instead,
#  we give general guidelines.
#
certificate_file = /usr/local/etc/raddb/certs/server.pem

#  Trusted Root CA list
#
#  This file can contain multiple CA certificates.
#  ALL of the CA's in this list will be trusted to
#  issue client certificates for authentication.
#
#  In general, you should use self-signed
#  certificates for 802.1x (EAP) authentication.
#  In that case, this CA file should contain
#  *one* CA certificate.
#
ca_file = /usr/local/etc/raddb/certs/ca.pem
```

FIGURE 3.17 – Implémentation du path du certificat racine

Les trois valeurs ci-dessous entre en compte dans la génération des fichiers aléatoires pour les échanges de clés.

```
#random_file = /dev/urandom
        #  This can never exceed the size of a RADIUS
        #  packet (4096 bytes), and is preferably half
        #  that, to accommodate other attributes in
        #  RADIUS packet.  On most APs the MAX packet
        #  length is configured between 1500 - 1600
        #  In these cases, fragment size should be
        #  1024 or less.
        #
#fragment_size = 1024

        #  include_length is a flag which is
        #  by default set to yes If set to
        #  yes, Total Length of the message is
        #  included in EVERY packet we send.
        #  If set to no, Total Length of the
        #  message is included ONLY in the
        #  First packet of a fragment series.
#include_length = yes
```

FIGURE 3.18 – Etat par défaut des valeurs de génération des fichiers aléatoires d'échange de clés

On décommente ces trois valeurs ci-dessus.

```
random_file = /dev/urandom
        #  This can never exceed the size of a RADIUS
        #  packet (4096 bytes), and is preferably half
        #  that, to accommodate other attributes in
        #  RADIUS packet.  On most APs the MAX packet
        #  length is configured between 1500 - 1600
        #  In these cases, fragment size should be
        #  1024 or less.
        #
fragment_size = 1024

        #  include_length is a flag which is
        #  by default set to yes If set to
        #  yes, Total Length of the message is
        #  included in EVERY packet we send.
        #  If set to no, Total Length of the
        #  message is included ONLY in the
        #  First packet of a fragment series.
include_length = yes
```

FIGURE 3.19 – Configuration des fichiers aléatoires d'échange de clés

### 3.2.5 Authentification du client

```
#        check_cert_cn = %{User-Name}
```

FIGURE 3.20 – Valeur d'authentification du client par défaut

```
check_cert_cn = %{User-Name}
```

FIGURE 3.21 – Valeur d'authentification du client décommenter

### 3.2.6 modification des fichiers "users" et "clients.cnf "

Dans le fichier "users" on renseigne les informations relatives au client qui pourra s'authentifier avec le serveur FreeRadius.

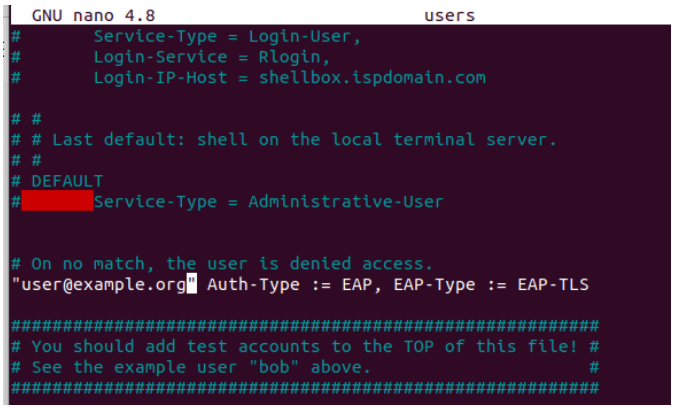



FIGURE 3.22 – Clients

Dans ce fichier, on renseigne les informations relatives au point d'accès .

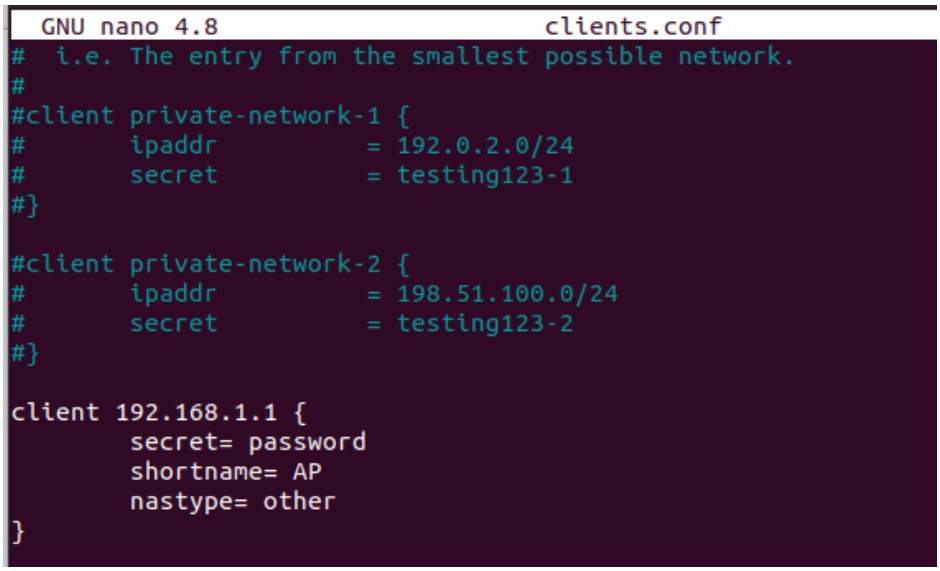



FIGURE 3.23 – Information du point d'accès

## 3.3 Tests

Pour les tests, nous allons utilise l'utilitaire eapol-test de wpa-supplicant.

### 3.3.1 wpa-supplicant

wpa-supplicant est une implémentation du composant WPA Supplicant, c'est-à-dire la partie qui s'exécute dans les stations clientes. Il implémente la négociation de clé WPA avec un authentificateur WPA et l'authentification EAP avec le serveur d'authentification. De plus, il contrôle l'itinérance et l'authentification/association IEEE 802.11 du pilote LAN sans fil.

### 3.3.2 eapol-test

eapol-test est un utilitaire distribué avec wpa-supplicant qui permet de tester des authentifications EAP. C'est un utilitaire très complet qui permet de vérifier les services RADIUS : méthode(s) EAP supportée(s), certificat électronique présenté, versions de TLS supportées, injection d'attributs RADIUS spécifiques... Passons à présent au téléchargement et à l'implémentation de eapol-test.

```
root@ubuntu:/home/ksroot# git clone --depth 1 --no-single-b
ranch https://github.com/FreeRADIUS/freeradius-server.git
Cloning into 'freeradius-server'...
remote: Enumerating objects: 33381, done.
remote: Counting objects: 100% (33381/33381), done.
remote: Compressing objects: 100% (19067/19067), done.
remote: Total 33381 (delta 22184), reused 21550 (delta 1367
2), pack-reused 0
Receiving objects: 100% (33381/33381), 23.85 MiB | 50.00 Ki
B/s, done.
Resolving deltas: 100% (22184/22184), done.
Updating files: 100% (4085/4085), done.
```

FIGURE 3.24 – Téléchargement du pacquage eapol-test

```
root@ubuntu:/home/ksroot/freeradius-server/scripts/ci# ./eapol_test-bui
ld.sh
Cloning into '/tmp/eapol_test.wt2Ow/hostapd'...
remote: Enumerating objects: 1437, done.
remote: Counting objects: 100% (1437/1437), done.
remote: Compressing objects: 100% (1320/1320), done.
remote: Total 1437 (delta 125), reused 552 (delta 60)
Receiving objects: 100% (1437/1437), 4.51 MiB | 4.00 KiB/s, done.
Resolving deltas: 100% (125/125), done.
Note: switching to '63962824309bb428e5f73d9caae08fcb949fbe36'.
```

```
make: Entering directory '/tmp/eapol_test.wt2Ow/hostapd/wpa_supplicant'
  CC  ../src/utils/wpabuf.c
  CC  ../src/utils/bitfield.c
  CC  ../src/utils/wpa_debug.c
  CC  eap_register.c
  CC  ../src/utils/common.c
  CC  bss.c
  CC  notify.c
  CC  ../src/utils/eloop.c
  CC  ../src/utils/os_unix.c
  CC  op_classes.c
  CC  wmm_ac.c
  CC  config.c
  CC  config_file.c
  CC  ../src/eap_peer/eap_tls.c
  CC  rrm.c
  CC  ../src/rsn_supp/preauth.c
  CC  ../src/rsn_supp/wpa_ie.c
```

FIGURE 3.25 – Configuration de eapol-test

Nous allons maintenant créer le fichier de configuration de EAP-TLS pour le test. Voici le contenu de ce fichier.

```
  GNU nano 4.8                  eap_tls.cnf
network={
        key_mgmt=WPA-EAP
        eap=TLS
        anonymous_identity="anonymous@<your-instiutions-domain>"

        # Uncomment to validate the server's certificate by checking
        # it was signed by this CA.
        ca_cert="raddb/certs/ca.pem"

        # supplicant's public cert
        client_cert="raddb/certs/client.crt"

        # supplicant's private key
        private_key="raddb/certs/client.key"

        # password to decrypt private key
        private_key_passwd="whatever"
}
```

## 3.4 Discussion

L'implémentation de EAP-TLS permet d'appliquer des signatures numériques afin de réaliser une authentification mutuelle entre le serveur et le client à travers l'échange de certificats suivant le protocole TLS. De plus, les communications entre ces deux entités sont chiffrées. Cependant, EAP-TLS, comme beaucoup de choses dans ce monde, n'est pas parfait .Ainsi, nous avons par exemple :

- sa complexité : L'utilisation de certificats accroît le nombre d'échanges nécessaires entre le Supplicant et le serveur d'authentification, ce qui peut accroître le temps nécessaire pour une connexion aboutie. La charge induite par les opérations de gestion de certificats est à prendre en compte. La distribution des certificats aux clients n'est pas non plus à négliger ;
- les attaques contre EAP : attaquer une session déjà établie, s'intercaler entre le système authentificateur et le client afin d'être authentifié à la place de ce dernier (attaque de type Man In the Middle).

## Conclusion

Au cours de ce chapitre, nous avons installé notre serveur d'authentification FreeRadius, généré les certificats client et serveur puis effectuer des tests avec eapol-test.

# Conclusion Générale

Au terme de notre étude, nous pouvons retenir que EAP-TLS basé sur l'architecture 802.1x est un ensemble de protocoles et de mécanisme qui permet d'identifier clairement les ordinateurs et équipements autorisés à accéder au réseau grâce à l'utilisation d'un serveur Radius et des certificats numériques ce qui donne plus de sureté à l'accès au réseau. La mise en place d'une telle solution a permi de sécuriser l'accès a un réseau sans fil afin d'en assurer la confidentialité et l'intégrité. Comme perspectives de ce projet, nous avons envisagé que EAP-TLS demande moins de ressources, permettant ainsi de faciliter son implémentation, ce qui augmentera certainement le nombre d'utilisateurs.

# References

[1] José DORDOIGNE *: Réseaux informatiques Notions fondamentales, Édition : ENI, mars 2015*

[2] Andrew TANENBAUM et David WETHERALL *: Réseaux, Édition : Dunod, septembre 2009*

[3] Aurélien GERON *: WiFi Professionnel- 3e édition - La norme 802.11, le déploiement, la sécurité ; Édition : Pearson Education, août 2011*

[4] Denis Bertrand *: Étude et mise en œuvre du protocole 802.1X dans le cadre de la politique de sécurité de Sphéria Valde France, 2015*

# Webographie


- Réseaux sans fils, https://www.cisco.com/c/fr_ca/solutions/small-business/resource-center/networking/wireless-network.html#~avantages consulté en mars 2022
- Architecture des réseaux sans fil, principes et. mise-en-œuvre , https://www.clicours.com/cours-informatique-architecture-des-reseaux-sans-fil-principes-et-mise-en-oeuvre/ consulté en mars 2022
- Protocole IEEE 802.11, https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/reseaux-locaux-42292210/protocole-ieee-802-11-te7379/norme-ieee-802-11-te7379v2niv10003.html#:~:text consulté en mars 2022
- Le Wifi, https://web.maths.unsw.edu.au/~lafaye/CCM/wifi/wifiintro.htm#:~:text consulté en mars 2022
- Norme wifi, https://www.planet-sansfil.com/dossiers/bien-choisir-sa-norme-wifi/ consulté en mars 2022
- FreeRadius, https://documentation.meraki.com/MR/Encryption_and_Authentication/Freeradius consulté en mai 2022
- FreeRADIUS, https://irp.nain-t.net/doku.php/310lansecure:30_radius:10_freeradius1 consulté en mai 2022
- , https://cric.grenoble.cnrs.fr/Administrateurs/Documentations/authentification/freeradius.php consulté en mai 2022
- , https://lists.freeradius.org/pipermail/freeradius-users/2016-August/084647.html consulté en mai 2022
- , https://www.securew2.com/blog/what-is-eap-tls consulté en mai 2022
- What is EAP-TLS? - SecureW2, consulté en mai 2022
- , https://learn.microsoft.com/fr-fr/troubleshoot/windows-server/networking/certificate-requirements-eap-tls-peap consulté en mai 2022

- Exigences relatives aux certificats lorsque vous utilisez EAP-TLS, https://learn.microsoft.com/fr-fr/troubleshoot/windows-server/networking/certificate-requirements-eap-tls-peap consulté en mai 2022
- What Is EAP-TLS, What Does It Do ? - JumpCloud, https://jumpcloud.com/blog/what-is-eap-tls consulté en mai 2022
- 802.1x EAP-TLS vs PEAP-EAP-TLS - Cisco Community , https://community.cisco.com/t5/network-access-control/802-1x-eap-tls-vs-peap-eap-tls/td-p/2901246#:~:text=Eap%2Dtls%20is%20based%20on,user%20sensitive%20information%20are%20encrypted. consulté en mai 2022
- eapoltest [Service mobilité pour la communauté RENATER] , https://www.eduroam.fr/eapol_test consulté en mai 2022
- eapoltest(8) — eapoltest — Debian testing, https://manpages.debian.org/testing/eapoltest/eapol_test.8.en.html consulté en mai 2022
- EAP Testing - Deploying RADIUS, http://deployingradius.com/scripts/eapol_test/ consulté en mai 2022

# Table des matières

**Dédicace** ii

**Remerciements** iii

**Résumé** iv

. . . . . iv

**Abstract** v

. . . . . v

**Liste des figures** vi

**Liste des acronymes** viii

**Introduction** 1

**1 Revue Littéraire** 3

Introduction . . . . . 3

1.1 Généralité sur les réseaux sans fil . . . . . 3

1.1.1 Historique et Définition . . . . . 3

1.1.1.1 Historique . . . . . 3

1.1.1.2 Définition . . . . . 4

1.1.2 Classification des réseaux sans fil . . . . . 4

1.1.3 Avantages et Inconvénients des réseaux sans fil . . . . . 5

1.1.3.1 Avantages . . . . . 5

1.1.3.2 Inconvénients . . . . . 5

1.2 IEEE 802.11 (Wi-Fi) . . . . . 6

1.2.1 Présentation . . . . . 6

1.2.2 Les différentes normes Wi-Fi . . . . . 6

1.2.3 Les équipements Wi-Fi . . . . . 7

1.2.3.1 Les adaptateurs sans fil ou carte d'accès . . . . . 7

1.2.3.2 Les points d'accès . . . . . 8

1.3 Architecture des réseaux Wi-Fi . . . . . 8

1.3.1 Le mode infrastructure . . . . . 8

1.3.2 Le mode Ad-hoc . . . . . 9

1.4 Sécurité des réseaux Wi-Fi . . . . . 9

1.4.1 La sécurité informatique et ses principes fondamentaux . . . . . 9

1.4.2 Les attaques contre les réseaux sans fil . . . . . 10

1.4.2.1 Denial of Service . . . . . 10

1.4.2.2 Le spoofing . . . 10
1.4.2.3 Le Men-In-The-Middle en milieu Wi-Fi . . . 10
1.4.3 Solutions de sécurisation des réseaux sans fil . . . 10
1.4.3.1 Le WEP . . . 10
1.4.3.2 le WPA . . . 10
1.4.3.3 L'architecture 802.1x . . . 11
Conclusion . . . 11

**2 Analyse choix technique et Conception** **12**
Introduction . . . 12
2.1 Généralités sur la solution choisie . . . 12
2.1.1 Choix de l'architecture de sécurisation . . . 12
2.1.2 Architecture du 802.1x . . . 12
2.1.3 Les méthodes d'authentification du 802.1x . . . 13
2.1.4 Les méthodes associées a EAP . . . 15
2.1.4.1 EAP-PAP . . . 15
2.1.4.2 EAP-MD5 . . . 16
2.1.4.3 EAP-TLS . . . 16
2.1.4.4 EAP-TTLS . . . 16
2.1.4.5 EAP-FAST . . . 16
2.1.4.6 EAP-SIM . . . 16
2.1.4.7 EAP-AKA . . . 16
2.1.4.8 EAP- LEAP . . . 17
2.1.4.9 EAP-PEAP . . . 17
2.2 Analyse du choix du type d'authentification EAP-TLS . . . 17
2.2.1 Le certificat numérique ou électronique . . . 18
2.2.2 Authentification par les certificats . . . 18
2.2.3 Déroulement de EAP-TLS . . . 18
Conclusion . . . 21

**3 Résultat obtenu et Perspectives** **22**
Introduction . . . 22
3.1 Installation de FreeRadius . . . 22
3.2 Implémentation de EAP-TLS . . . 24
3.2.1 Génération des différents certificats . . . 24
3.2.2 Génération des paramètres de l'algorithme de Diffie-Hellman . . . 28
3.2.3 Modification du fichier de configuration par défaut du protocole EAP . . . 28
3.2.4 Implémentation des chemins relatifs aux certificats dans le fichier de EAP . . . 28
3.2.5 Authentification du client . . . 29
3.2.6 modification des fichiers "users" et "clients.cnf " . . . 29
3.3 Tests . . . 30
3.3.1 wpa-supplicant . . . 30
3.3.2 eapol-test . . . 30
3.4 Discussion . . . 32
Conclusion . . . 32

Conclusion 33

References 34

Webographie 35

Table des matières 37