

Université Claude Bernard Lyon 1  
Master Systèmes Informatique et Réseaux  
2006/2007



Travail d'Etude et de Recherche

## **Services d'authentification et annuaires**

Thibaut DE DOMPSURE, Vincent SAGE

12 décembre 2006

## Table des Matières

<b>1- Introduction</b> .....	3
<b>2- Protocoles d'authentications.</b> .....	4
2.1- PAP.....	4
2.2- CHAP .....	4
2.3- MS-CHAP.....	5
2.3.1-Version 1 .....	5
2.3.2-Version 2 .....	5
2.4- EAP .....	6
2.4.1-EAP-MD5.....	7
2.4.2-EAP-TLS.....	7
2.4.3-EAP-TTLS / PEAP .....	7
2.4.4-LEAP .....	7
2.5- 802.1X.....	7
<b>3- Services d'authentification de la couche réseau</b> .....	8
3.1- IPsec .....	8
3.1.1-Protocole AH .....	9
3.1.2-Protocole ESP .....	10
3.2- NuFW .....	11
<b>4- Services d'authentification de la couche session</b> .....	13
4.1- SSL/TLS .....	13
<b>5- Services d'authentification applicatifs</b> .....	14
5.1- RADIUS .....	14
5.2- TACACS+ .....	16
5.3- Kerberos .....	17
5.4- Authentification web .....	19
5.4.1-PHP/ASP.....	19
5.4.2-.htaccess.....	19
<b>6- Annuaires</b> .....	20
6.1- Active Directory .....	20
6.2- LDAP.....	20
6.3- NIS/NIS+ .....	21
<b>7- Les évolutions à venir</b> .....	22
7.1- Diameter .....	22
7.2- Liberty Alliance .....	22
7.3- La biométrie .....	23
<b>8- Conclusion</b> .....	24
<b>9- Bibliographie</b> .....	25
9.1- Documentations écrites .....	25
9.2- Documentations en lignes .....	25

## 1- Introduction

Sécuriser le système d'information est de plus en plus difficile, surtout à l'heure où le nombre d'applications et le degré d'ouverture vers l'extérieur vont croissant. Définir les personnes autorisées à accéder au système d'information constitue l'une des bases de la sécurité. Cette authentification des utilisateurs qu'elle soit interne ou externe au cercle de confiance est indispensable.

L'authentification englobe souvent des concepts et des approches différentes.

Il y a plusieurs moyens d'authentification. Ils sont généralement regroupés en trois grandes catégories :

- Il y a ce que l'on connaît : un mot de passe, un code PIN ...
- Il y a ce que l'on a : une carte à puce, un certificat ...
- Il y a ce que l'on est : la biométrie.

L'authentification par mot de passe est utilisée en grande majorité dans les systèmes sécurisés, car c'est la plus simple à mettre en place. Nous verrons néanmoins les autres possibilités d'authentifications et les avantages.

Sachant que les services définissent l'ensemble des opérations accessibles par des protocoles, les services d'authentifications regroupent donc l'ensemble des opérations pour s'authentifier, quel que soit le protocole et quelle que soit la couche ISO.

Les services d'authentification ont tous besoins d'une « base de connaissance » afin d'y stocker toutes sortes d'informations. Ces informations peuvent être très variées et peuvent établir un véritable profil en fonction de l'entité qui s'authentifie. Cette « base de connaissance » est appelée « annuaire ». L'annuaire est donc aussi un service permettant de maintenir de façon cohérente, contrôlé, organisé, une grande quantité de donnée relative à des personnes, à des machines ou des ressources.

Notre étude bibliographique porte sur les services d'authentifications et les annuaires.

## 2- Protocoles d'authentications.

Les mécanismes d'authentification décrits dans cette partie ont tout d'abord été des protocoles de couche 2 (liaison de données) puisqu'ils ont été initialement utilisés par le protocole PPP (*Point-to-Point Protocol*) qui permet l'ouverture de session sur le réseau RTC. Actuellement, ils sont également utilisés dans la couche réseau grâce aux évolutions de PPP : PPPoA (over ATM) et PPPoE (over Ethernet) qui sont principalement utilisés pour ouvrir des connexions ADSL. Cependant, ces mécanismes sont les briques de nombreux serveurs et applications d'authentification comme Radius.

### 2.1- PAP

Le protocole PAP (*Password Authentication Protocol*), utilisé avec le protocole PPP, permet d'identifier un utilisateur auprès d'un serveur PPP en vue d'une ouverture de connexion sur le réseau.

Après une phase de synchronisation entre le client et le serveur pour définir l'utilisation du protocole PPP et PAP, le processus d'authentification se fait en deux étapes :

- Le client envoie son nom PAP ainsi que son mot de passe en clair.
- Le serveur qui détient une table de noms d'utilisateurs et de mots de passe vérifie que le mot de passe correspond bien à l'utilisateur et valide ou rejette la connexion.



Fig. 1 : Illustration des 2 étapes d'authentification du protocole PAP

PAP est le plus simple des protocoles d'authentification, il est donc très facile à implémenter. Mais étant donné que le mot de passe circule en clair sur le réseau, c'est aussi le moins sécurisé et il est donc fortement déconseillé car il ne procure aucune sécurité. D'autre part, même si le mot de passe est crypté, il est toujours possible d'utiliser un sniffer afin de capturer la requête d'authentification et de la réutiliser pour s'authentifier (attaque par replay).

### 2.2- CHAP

Contrairement au protocole PAP, le protocole CHAP (*Challenge Handshake Authentication Protocol*) permet une authentification sécurisée par hachage MD5 (*Message Digest 5*). MD5 est une fonction de hachage cryptographique permettant d'obtenir l'empreinte numérique d'un message à partir de laquelle il est impossible de retrouver le message original. Ainsi, en envoyant l'empreinte du mot de passe au serveur, le client peut montrer qu'il connaît bien le mot de passe sans avoir à réellement l'envoyer sur le réseau.

Après le même type de synchronisation que pour le protocole PAP, le mécanisme d'authentification est basé sur un « défi » (*challenge*) en 3 étapes :

- Le serveur envoie au client un nombre aléatoire de 16bits ainsi qu'un compteur incrémenté à chaque envoi.
- Le client génère une empreinte MD5 de l'ensemble constitué par : son mot de passe, le nombre aléatoire et le compteur reçu puis il envoie cette empreinte.
- Le serveur calcul également de son côté l'empreinte MD5 grâce au mot de passe du client stocké localement puis il compare son résultat à l'empreinte envoyée par le client. Si les deux empreintes sont identiques, le client est bien identifié et la connexion peut s'effectuer sinon, elle est rejetée.

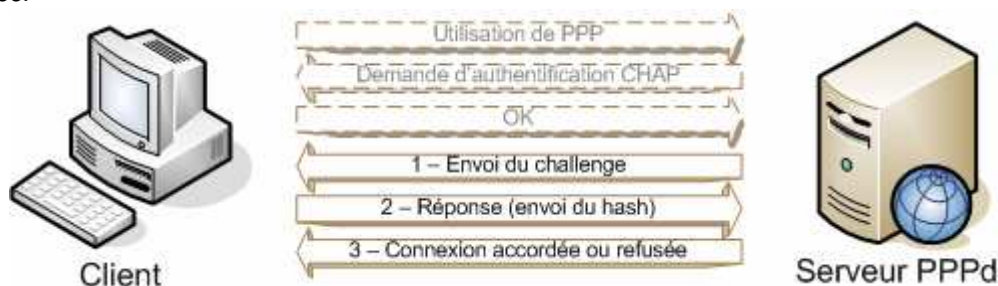


Fig. 2 : Illustration des 3 étapes d'authentification du protocole CHAP

Ce mécanisme d'authentification procure à CHAP deux avantages : tout d'abord, si la requête d'authentification envoyée par le client est interceptée, elle ne pourra pas être rejouée, en effet chaque empreinte calculée par le client est unique puisqu'elle fait intervenir le compteur qui est incrémenté à chaque envoi par le serveur. D'autre part, lors d'une session établie par le protocole CHAP, le serveur envoie régulièrement des *challenges* au client de façon à vérifier son identité, cette mesure de protection supplémentaire permet donc de se prémunir des vols de session.

## 2.3- MS-CHAP

### 2.3.1- Version 1

MS-CHAP (*Microsoft Challenge Handshake Authentication Protocol*) est la version spécifique de CHAP mise au point par Microsoft. Plus qu'une simple version prioritaire, MS-CHAP apporte également quelques améliorations à CHAP. Un des principaux inconvénients de CHAP est que le serveur doit détenir les mots de passe des utilisateurs en clair pour pouvoir vérifier l'empreinte MD5 envoyée par les clients, ce qui constitue une vulnérabilité potentielle en cas de compromission du serveur. Pour remédier à cette faiblesse, le protocole MS-CHAP intègre une fonction de hachage propriétaire permettant de stocker sur le serveur un hash intermédiaire du mot de passe. Ainsi, en travaillant uniquement avec ce hash intermédiaire à la place du mot de passe, le client et le serveur peuvent réaliser le même type de procédure que dans la figure 2, ainsi, le mot de passe en clair n'a plus besoin d'être stocké sur le serveur.

### 2.3.2- Version 2

Malgré l'avancée de MS-CHAP par rapport à CHAP, Microsoft créa une seconde version du protocole (MS-CHAP-v2) pour résoudre deux principales faiblesses de MS-CHAP-v1, d'une part le fait que le client ne puisse pas vérifier l'authenticité du serveur sur lequel il veut se connecter et d'autre part que l'algorithme de hachage propriétaire utilisé soit très vulnérable à des attaques par *brute-force*.

Voici le fonctionnement du processus d'authentification mutuelle fourni par MS-CHAP-v2 :

- Le serveur d'accès distant envoie une demande de vérification au client contenant un identificateur de session I et une chaîne C1 générée aléatoirement.
- Le client envoie alors une réponse contenant : son nom d'utilisateur, une chaîne aléatoire C2 et un hash de l'ensemble formé par 3 éléments : la chaîne C1, l'identificateur de session I et son mot de passe.

- Le serveur vérifie la réponse du client et il renvoie une réponse contenant : une chaîne indiquant le succès ou l'échec de l'authentification, et un hash de l'ensemble formé par 3 éléments : la chaîne C2, l'identificateur de session I et son mot de passe.
- Le client vérifie à son tour la réponse d'authentification et établit la connexion en cas de réussite.

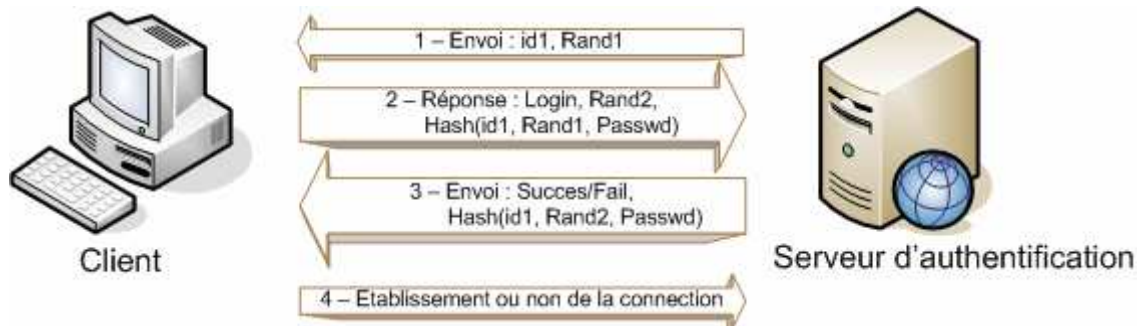


Fig. 3 : Illustration des 4 étapes d'authentification du protocole MS-CHAP-v2

Cette méthode d'authentification est bien mutuelle car elle permet effectivement au client d'être sûr de l'identité du serveur car seul le serveur peut lui renvoyer son mot de passe dans le hash à l'étape 3. Du fait de cette amélioration notable et des faiblesses de MS-CHAP-v1, seule la seconde version de MS-CHAP est supportée par le dernier système d'exploitation de Microsoft : Windows Vista.

## 2.4- EAP

EAP (*Extensible Authentication Protocol*) n'est pas directement un mécanisme d'authentification comme peuvent l'être PAP ou CHAP, il s'agit en réalité d'une extension du protocole PPP qui a permis d'universaliser et de simplifier l'utilisation des différents protocoles dans le cadre des réseaux sans fils et les liaisons Point-A-Point. EAP contient une douzaine de méthodes d'authentification, les plus utilisées étant EAP-MD5, EAP-TLS, EAP-TTLS, LEAP ou encore EAP-AKA pour l'UMTS.

Il faut distinguer des types de trafic EAP : celui entre le client et le point d'accès : EAP over LAN (utilisant un média 802.11a, b ou g) et celui entre le point d'accès et le serveur d'authentification : EAP over Radius.

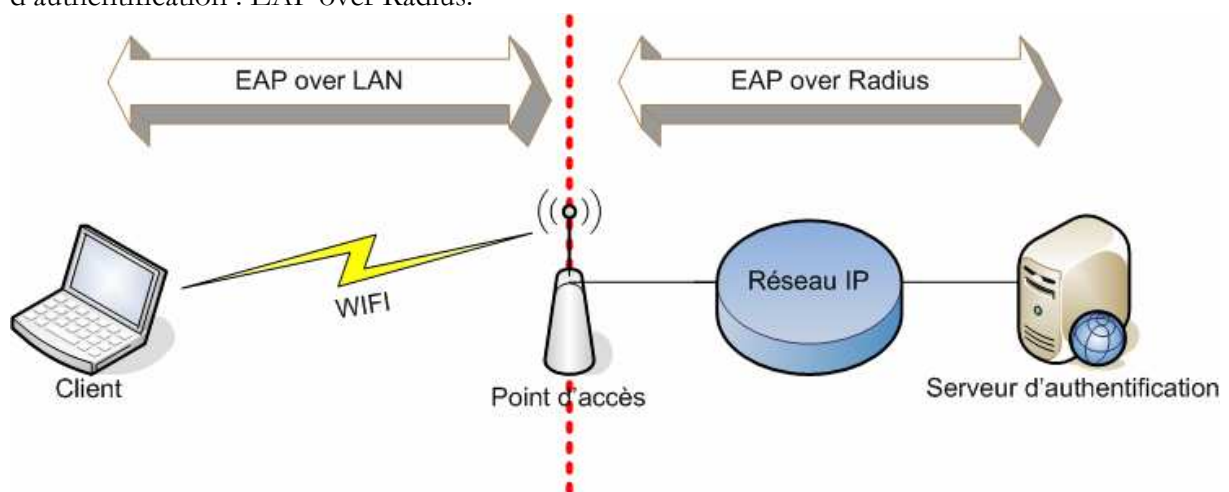


Fig. 4 : Différents types de trafic EAP

### 2.4.1- EAP-MD5

EAP-MD5 est une des méthodes d'authentification les plus simples d'EAP, il est très simple à mettre en place et son mécanisme d'authentification est semblable à la méthode CHAP (fig. 2). Cependant, cette méthode ne permet pas la distribution de clés WEP, il est donc inutilisable pour les réseaux sans fils. De plus, cette méthode ne permet donc pas d'authentification mutuelle et tout comme CHAP, elle est vulnérable aux attaques par force-brute ou au vol de session.

### 2.4.2- EAP-TLS

EAP-TLS est basé sur le protocole TLS que nous allons développer un peu plus loin. Ce protocole utilise deux certificats permettant une authentification mutuelle afin de créer un canal sécurisé. De cette façon, même si le mot de passe client est découvert, il ne sera d'aucune utilité sans le certificat client. EAP-TLS offre donc un très bon niveau de sécurité et est un standard ouvert IETF, pour ces raisons, il est implanté dans de très nombreux matériels sans fils.

Le principal reproche que l'on peut faire à ce protocole est, qu'étant donné l'obligation d'utiliser un certificat différent pour chaque client, il peut s'avérer compliqué et coûteux à mettre en place dans un grand parc de machines.

### 2.4.3- EAP-TTLS / PEAP

Assez similaire avec EAP-TLS, cette méthode propose également une authentification mutuelle à la différence que le client n'a pas besoin de détenir un certificat. En effet, dans un premier temps le client identifie bien le serveur en utilisant un certificat (validé par une autorité de certification), mais ensuite, le serveur identifie le client par un couple login, password. A partir du moment où le serveur est identifié, un tunnel TLS chiffré est établi permettant au client d'envoyer son login et mot de passe via les méthodes vue ci-avant : PAP, CHAP ou MS-CHAPv2. L'avantage d'EAP-TTLS est donc de permettre une authentification mutuelle sans avoir recours à une infrastructure de gestion de clés (IGC).

Le protocole PEAP (*Protected EAP*) fonctionne de la même façon qu'EAP-TTLS à la seule différence qu'il ne peut fonctionner qu'en direct avec un serveur radius supportant EAP, alors que EAP-TLS est plus souple et peut utiliser un serveur intermédiaire TTLS redirigeant les messages vers un serveur Radius non EAP.

### 2.4.4- LEAP

LEAP (*Lightweight EAP*) est une implémentation propriétaire d'EAP développée par Cisco sur ses équipements actifs. Ce protocole fonctionnant avec des cartes à puces présente plusieurs défaillances puisque d'une part il utilise la méthode MS-CHAPv1 pour établir l'identification du client par le point d'accès et d'autre le login/mot de passe en clair sur le réseau.

## 2.5- 802.1X

802.1X n'est pas réellement un protocole mais une norme définissant un contrôle de l'accès au réseau basé sur le port. Ce standard mis au point par l'IEEE (*Institute of Electrical and Electronics Engineers*) a pour objectif de réaliser une authentification au moment de la connexion au réseau (filaire ou sans fil) pour permettre la connexion à ce dernier. Pour réaliser l'authentification, les échanges dans 802.1X sont basés sur EAP (voir ci-avant).

802.1X définit trois types d'éléments :

- Le demandeur (*supplicant*) qui représente l'hôte désirant accéder au réseau

- L'authentificateur direct (*authenticator*) : C'est un élément actif qui met en œuvre le processus d'ouverture ou de fermeture de l'accès au réseau en fonction de la réponse du serveur d'authentification. L'authentificateur peut aussi bien être un commutateur, un point d'accès sans fils ou un commutateur/routeur.

- Le serveur d'authentification : il répond aux requêtes de l'authentificateur en lui indiquant si le demandeur peut se connecter ou pas.

Le serveur d'authentification peut très bien être intégré au commutateur mais, dans ce cas, il ne permet pas une gestion centralisée des accès. C'est pourquoi là plupart du temps, il d'agit d'un serveur indépendant, comme Radius.

802.1X définit deux types des ports sur l'authentificateur les ports contrôlés et non contrôlés :

- Les ports non contrôlés permettent à l'authentificateur de communiquer avec d'autres nœuds du réseau. Le port utilisé pour communiquer avec le serveur d'authentification est donc non contrôlé.

- Les ports contrôlés permettent au demandeur de communiquer avec le réseau après avoir été autorisé. Avant cette autorisation, aucune trame ne peut être émise du demandeur vers le réseau.

L'ensemble de ces deux ports s'appelle « le point d'accès au réseau » (PAE), il est indépendant de la nature physique de ces ports (RJ45, connecteur SC, MT-RJ, 802.11a,b,g).

Si une authentification est refusée par le serveur, un système de temporisation au niveau de l'authentificateur est effectué (60 secondes par défaut) permettant de limiter les attaques de type « force brute ».

Malgré un contrôle d'accès efficace au réseau, 802.1X possède néanmoins quelques faiblesses : par exemple s'il n'y a aucune restriction sur le nombre d'adresses MAC autorisées par port, il est tout à fait possible pour un utilisateur de brancher un hub et donc de faire bénéficier à d'autres personnes le port ouvert. D'autre par, des attaques par rejeu et des vols de session sont possibles, particulièrement lors de son utilisation en milieu Wifi.

802.1X propose donc un niveau de sécurité correct mais pour une protection optimale il doit être couplé à d'autres technologies (par exemple de chiffrement), particulièrement en Wifi avec WPA.

### 3- Services d'authentification de la couche réseau

#### 3.1- IPsec

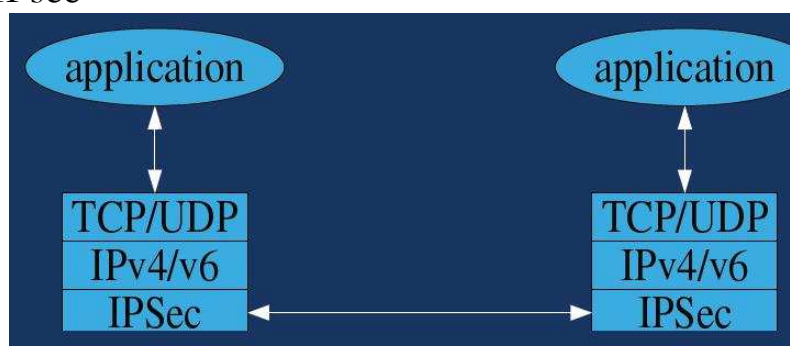


Fig. 5 : La pile TCP/IP et IPsec

Profitant de la définition du nouveau protocole IPv6, l'IAB (*Internet Architecture Board*) a décidé d'intégrer des services de sécurité dans le protocole IP lui-même. IPsec s'insère dans la pile de protocole TCP/IP, au niveau IP. Le niveau de sécurité d'IPsec est lié à la puissance des algorithmes utilisés.

Avant de pouvoir détailler le service d'authentification, il est nécessaire de connaître la différence entre le mode de transport et le mode tunnel.

**Deux modes peuvent être utilisés :**

- Le mode transport : il est plutôt utilisé dans une configuration point à point. Il prend le flux de la couche 4 du modèle OSI, réalise les mécanismes de signature et de chiffrement, puis transmet les données à la couche IP. Les en-têtes du message sont donc construits par le protocole IP. Les adresses d'émission et de destination des paquets IP sont envoyées en clair.
- Le mode tunnel : il est utilisé entre deux passerelles de sécurité pour relier des systèmes d'informations d'entreprises. Il chiffre les flux jusqu'à la couche IP incluses, ce qui offre un niveau de sécurité supérieur au mode transport.

Notre but ici ne sera pas de détailler le fonctionnement du protocole IPsec mais, de voir plus particulièrement, le service d'authentification offert par IPsec. C'est une authentification de couche 3 : elle authentifie des hôtes et non des personnes. L'implémentation de ce service se traduit par des informations supplémentaires se trouvant dans les en-têtes, en fonction du mode d'acheminement des données et des modes utilisés.

Dans tous les cas, les données sont authentifiées par le champ appelé ICV (*Integrity Check Value*). Ce sont des données d'authentification. Ce champ est de taille variable. C'est un terme générique qui désigne soit une signature numérique, soit un code d'authentification de message (*Message Authentication Mac*). AH et ESP ne spécifient pas d'algorithme de signature particulier, ceux-ci sont décrits séparément, cependant la RFC 2402 et 2406 demandent à ce que les algorithmes de hachage MD5 et SHA-1 soient implémentés. Le résultat obtenu est une empreinte fixe de données :

- Si c'est un code d'authentification de message, l'ICV est le résultat d'un hachage à sens unique à clé secrète. L'empreinte dépend des données et de la clé.
- Si c'est une signature numérique, l'ICV est en fait le chiffrement d'une empreinte à partir d'une clé publique. Cette technique a l'avantage d'ajouter le service de non répudiation en plus d'assurer l'authentification et l'intégrité. Mais elle est cependant moins utilisée en pratique car une cryptographie à clé publique consomme plus de ressource

Le service d'authentification ne peut-être rendu que s'il y a eu, au préalable, un dialogue initial permettant l'échange de clés. Il peut être fait manuellement s'il y a peu de machine. Si le nombre de machines augmente, la spécification IPsec propose des procédés d'échange automatique de clés comme le protocole IKE (*Internet Key Exchange – RFC 2409*).

Deux protocoles sont proposés par la norme : AH et ESP. On les utilisent en fonction des services que l'on désire avoir.

### 3.1.1- Protocole AH

Le protocole AH (*Authentication Header – RFC2402*) : garanti l'authentification de la provenance du paquet (en-tête IP), l'intégrité et le non-rejeu des données.

#### **AH dans le mode transport**

Dans le mode transport, l'intégrité et l'authentification sont assurées par l'insertion de l'en-tête AH entre l'en-tête IP d'origine et la charge utile IP (les données). La valeur ICV qui se trouve dans l'en-tête AH est calculée sur base de l'en-tête IP d'origine, de l'en-tête AH et de la charge utile IP.

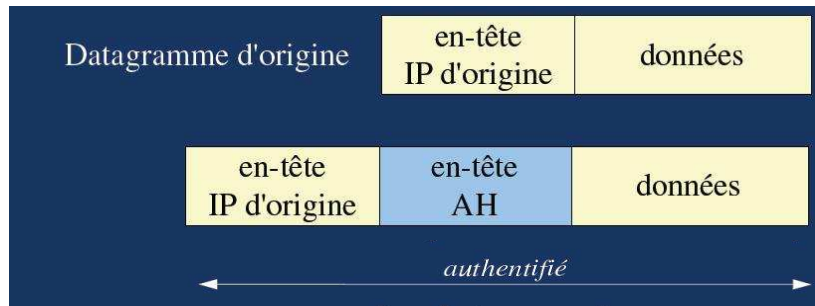


Fig. 6 : Datagramme AH

**AH dans le mode tunnel**

Dans le mode tunnel, AH encapsule un paquet IP, avec un en-tête AH et un nouvel en-tête IP, puis signe ce nouveau paquet en vue d'en garantir l'intégrité et l'authentification. La valeur ICV se trouve dans l'en-tête d'authentification.

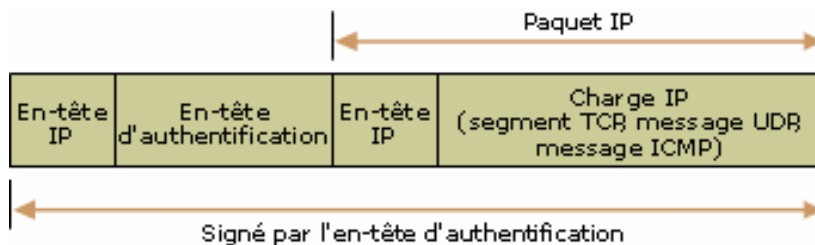


Fig. 7 : Datagramme AH (mode tunnel)

Ainsi, on est sûr que les datagrammes IP reçus sont bien émis par l'hôte dont l'adresse IP est indiquée comme adresse source dans les en-têtes.

On peut remarquer que pour le protocole AH, l'adresse d'origine et l'adresse destination font partie du contrôle d'intégrité. Par conséquent, une translation d'adresse (NAT) est impossible.

3.1.2- Protocole ESP

Le protocole ESP (*Encapsulated Security Payload* – RFC 2406) : garanti l'authentification de la charge utile d'IP (segment TCP, message UDP, message ICMP...), l'intégrité, le non-rejeu des données et la confidentialité des données.

ESP peut être utilisé seul ou combiné à AH.

**ESP avec le mode transport**

En mode de transport, ESP ne signe pas l'ensemble du paquet. Seule la charge utile IP (les données) est protégée, et non l'en-tête IP. La valeur ICV se trouve dans les données d'authentification et elle est calculée sur base de l'en-tête ESP, des données de la charge utile et du code de fin ESP (remorque ESP sur le schéma).

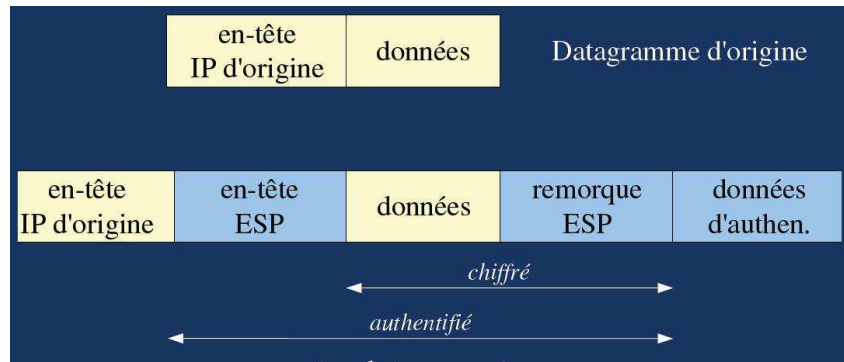


Fig. 8 : Datagramme ESP

**ESP avec le mode tunnel**

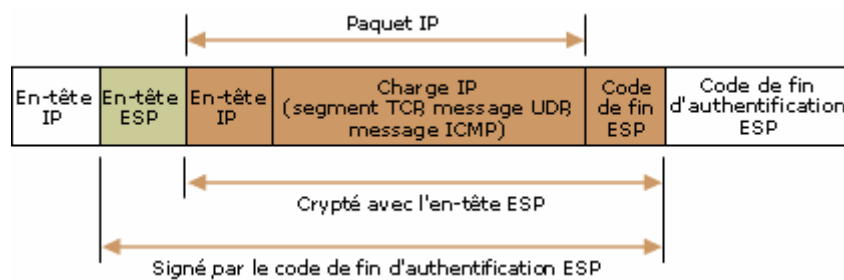


Fig. 9 : Datagramme IP (mode transport)

En mode tunnel, ESP encapsule un paquet IP avec un en-tête ESP et un nouvel en-tête IP, ainsi qu'un code de fin d'authentification ESP où se trouve la valeur ICV. On peut remarquer que l'en-tête IP n'est pas protégé par le hachage. L'en-tête IP n'est donc pas nécessairement protégé contre une modification.

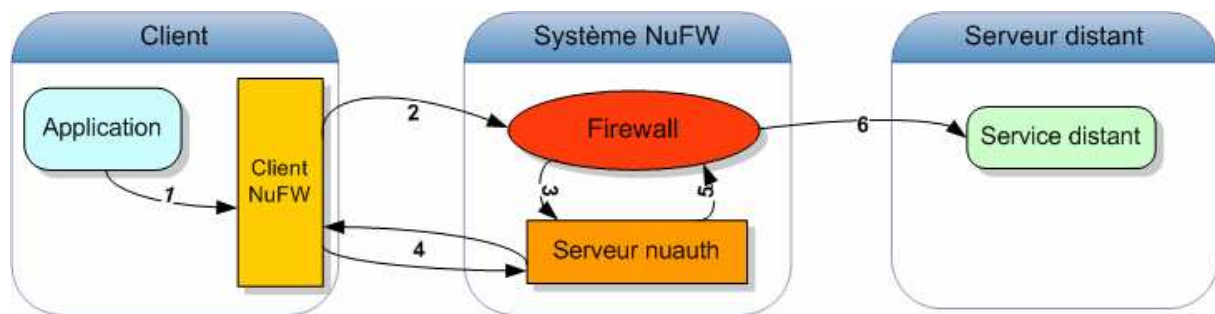
IPsec est un protocole de niveau 3 et il ne fournit donc qu'une authentification de niveau égal, c'est-à-dire une authentification des machines. L'utilisation du protocole AH ou ESP n'offre pas le même niveau d'authentification. Il faut utiliser ESP et AH pour assurer le chiffrement, l'intégrité des données et l'authentification de l'en-tête IP.

**3.2- NuFW**

NuFW est un pare-feu authentifiant ouvert et libre (licence GPL v2) disponible sous linux et basé sur Netfilter. Comme tout pare-feu basé sur Netfilter, NuFW permet le filtrage et le routage des paquets, mais c'est uniquement à son rôle de passerelle d'authentification que nous allons nous intéresser.

Concrètement, NuFW fonctionne comme une passerelle par laquelle passent toutes les requêtes des clients, mais contrairement à un simple filtrage au niveau IP, NuFW prend réellement en compte la notion d'utilisateur en interrogeant le client nufw grâce au serveur d'authentification "nuauth". Ainsi, cette passerelle peut autoriser ou nom la connexion en fonction des permissions de l'utilisateur (stockées dans un LDAP par exemple) et mettre à jour une table des connexions authentifiées.

Le principe de NuFW est d'authentifier et d'associer un identifiant à chaque paquet émis par un client suivant l'algorithme suivant :

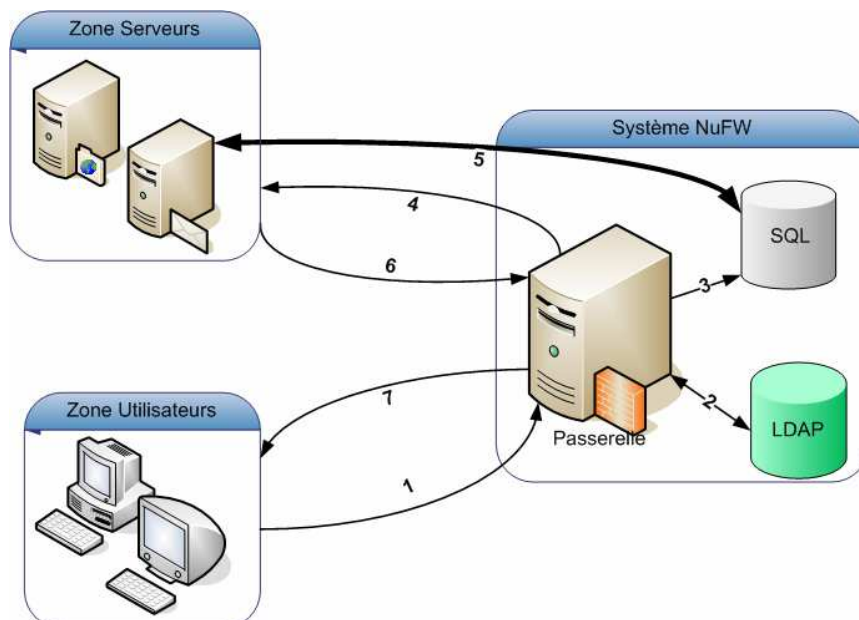


**Fig. 10:** Algorithme d'authentification d'un paquet

- 1- Une application classique située sur le client envoie un paquet.
- 2- Le client NuFW envoie une requête au serveur NuFW
- 3- Le serveur NuFW envoie une demande d'authentification au serveur d'authentification nuauth
- 4- Le serveur nuauth cherche les permissions en interrogeant le client NuFW .
- 5- Nuauth envoie la réponse au serveur NuFW.
- 6- En fonction de la réponse, le serveur NuFW autorise ou non la transmission du paquet au serveur distant.

NuFW possède une option permettant de mettre en place très facilement une solution d'authentification unique (SSO : *Single Sign On*). Cette solution SSO permet à l'utilisateur de s'authentifier une seule fois au niveau de la passerelle, les authentifications étant effectuées par les serveurs qui vont se charger d'interroger le Système NuFW. En activant cette option, NuFW stocke dans une base SQL les paramètres IP d'une connexion, l'utilisateur qui l'a ouverte et l'état de la connexion (fermée, établie, ouverte). Ainsi, comme tout serveur distant connaît les paramètres IP des connexions entrantes, il peut très facilement faire le lien avec l'utilisateur en interrogeant la base SQL du serveur NuFW et autoriser ou non l'accès.

Voici le schéma de fonctionnement de la solution SSO de NuFW



**Fig. 11:** Fonctionnement du SSO NuFW

- 1- L'utilisateur envoie un paquet à la passerelle pour accéder à un serveur distant.
- 2- La passerelle NuFW vérifie l'autorisation de passage du paquet (voir ci-avant).
- 3- NuFW remplit la table SQL avec l'IP, l'utilisateur et l'état de la connexion.
- 4- La passerelle NuFW transmet le paquet au serveur distant.

- 5- Le serveur interroge la base SQL pour connaître l'utilisateur qui a émis le paquet.
- 6/7- En fonction de l'utilisateur l'application répond à la passerelle qui transmet au client.

## 4- Services d'authentification de la couche session

### 4.1- SSL/TLS

Le protocole propriétaire SSL (*Secure Socket Layer*) a été développé par Netscape pour sécuriser les transactions entre un client et un serveur web (généralement un site marchand). Il existe plusieurs versions de SSL : la version 3.0 est la plus répandue, mais depuis la version 3.1 SSL a été rebaptisé TLS version 1 (*Transport Layer Security*) et standardisée par l'IETF. SSL ou TLS sont des protocoles situés dans la couche session et fonctionnent donc indépendamment des applications qui les utilisent. Ils permettent d'assurer non seulement des services d'authentification, mais aussi des services de confidentialité par chiffrement symétrique (DES, IDEA, 3DES, ...) et d'intégrité par hachage MD5 ou SHA-1. Actuellement SSL et TLS sont principalement utilisés pour sécuriser des échanges HTTP, ce qui donne le protocole HTTPS, mais en réalité, SSL et TLS permettent de créer des tunnels sécurisés et peuvent donc être utilisés avec n'importe quel protocole sécurisé de couche supérieure comme FTP, SMTP, etc. De plus, les nombreuses bibliothèques disponibles permettent d'implémenter SSL/TLS dans de nombreux langages (PHP, C++, etc.).

Les deux protocoles ne sont pas interopérables mais TLS propose un mécanisme de compatibilité ascendante avec SSL. TLS n'apporte pour l'instant que peu d'améliorations par rapport à SSL, mais dans l'avenir, les nouveaux algorithmes de chiffrement lui seront intégrés et il deviendra plus sécurisé que SSL.

SSL/TLS, contrairement aux protocoles que nous avons définis jusqu'à maintenant ne sert pas à authentifier un client auprès d'un serveur, mais un serveur auprès d'un client afin que le client soit sûr qu'il ne communique pas avec un serveur factice. Une fois cette authentification réalisée, le client et le serveur s'échangent une clé secrète qui va leur permettre de communiquer en chiffrant leurs messages avec un algorithme de chiffrement symétrique. La phase de communication chiffrée s'appelle : SSL Record Protocol et la phase d'authentification : SSL Handshake Protocol. Dans le cadre de notre recherche, c'est cette négociation qui nous intéresse, voici ses différentes étapes :

- 1- Le client se connecte au serveur puis lui envoie un message CLIENT-HELLO contenant plusieurs informations dont la version de SSL/TLS et les algorithmes de chiffrements qu'il supporte.
- 2- Le serveur lui renvoie le nom de l'algorithme le plus sûr qu'il a en commun avec le client et un certificat X509 contenant sa clé publique. Ce certificat est signé par une autorité de certification.
- 3- A la réception du certificat, le client doit effectuer trois vérifications :
  - La période de validité du certificat X509 : généralement un certificat est valide un an, si la date et l'heure ne sont pas dans la période de validité le processus d'authentification ne va pas plus loin.
  - La validité de l'autorité de certification (AC) : chaque client SSL possède une liste de certificats des autorités de certifications de confiance, si le nom distinct (DN : *Distinguished Name*) de l'AC correspond bien au DN d'une AC de confiance existante dans la liste alors elle est validée.
  - La validité de la signature numérique de l'émetteur : En utilisant la clé publique de l'AC, le client peut vérifier la signature du certificat X509 du serveur, en effet, cette signature a été réalisée par la clé privée de l'AC donc seule l'AC a pu signer le certificat.Une quatrième étape optionnelle peut être réalisée pour vérifier que l'IP du serveur correspond bien à celle définie dans le certificat.
- 4- Une fois le serveur authentifié, le client calcule et envoie un code secret intermédiaire qu'il chiffre avec la clé publique RSA du serveur.

- 5- Le serveur déchiffre le code secret intermédiaire avec sa clé privée RSA. Le client et le serveur utilisent ce code secret intermédiaire pour créer une clé secrète de session compatible avec l'algorithme de chiffrement symétrique définit à l'étape 2.
- 6- Le serveur envoie au client un message chiffré avec la clé secrète de session indiquant au client que la négociation est terminée et que les prochains échanges seront chiffrés avec cette clé secrète de session. TLS propose également une authentification du client auprès du serveur grâce à un certificat client dans ce cas, à l'étape 4 le client envoie également son certificat au serveur pour que celui-ci puisse vérifier sa validité.

SSL/TLS propose donc une authentification forte basée sur des clés RSA très difficilement cassables, cependant toute la force de ce protocole repose uniquement sur la validation certificat, c'est pour cette raison que SSL peut malgré tout être vulnérable à une attaque du type « Man in the middle ». En effet, si quelqu'un arrive à intercepter toutes les communications entre le client et le serveur, il peut alors récupérer le certificat légitime envoyé par le serveur et lui substituer son propre certificat valide de cette façon le client valide automatiquement le certificat et le pirate peut à se faire passer pour le serveur auprès du client et pour le client et auprès du serveur. Le pirate peut ainsi déchiffrer toutes les communications de façon totalement transparente.

## 5- Services d'authentification applicatifs

### 5.1- RADIUS

*Remote Authentication Dial-In User Service* - RFC 2865 et RFC 2866  
RADIUS est une norme de l'IETF (*Internet Engineering Task Force*).

C'est un protocole d'authentification standard Client/serveur qui permet de centraliser les données d'authentification : Les politiques d'autorisation et de droits d'accès, traçabilité. Ce processus doit être relié à une source d'informations, qui est souvent un annuaire LDAP.

Auparavant, les noms et les mots de passe des utilisateurs devaient être dupliqués sur chaque serveur pouvant être accédé à distance (par un modem RTC par exemple). L'arrivée de RADIUS permet aux fournisseurs d'accès Internet d'authentifier les utilisateurs distants connectés, à partir d'une seule base utilisateurs. Ce protocole avait particulièrement un sens avant l'ADSL illimité, car il permettait de mesurer le temps précis de connexion des abonnés et facturer en conséquence.

L'identification sur les sites Web peut aussi être gérée par RADIUS. Apache est sans doute le client RADIUS le plus répandu (le module `mod_auth_radius` permet à Apache de valider une authentification en interrogeant un serveur Radius). Aujourd'hui, ce protocole est aussi souvent utilisé pour les connexions à Internet sans fil (WLAN - avec le protocole 802.1X qui assure l'identification par port pour l'accès à un réseau). On le retrouve aussi au sein de la téléphonie sur IP comme outil de gestion des connexions, autour du protocole SIP notamment. Dans ce cas, l'annuaire SIP chargé de l'authentification communique avec le serveur Radius en utilisant ce protocole.

Son fonctionnement est simple :

1. L'utilisateur (le supplicand dans la RFC) se connecte (via PPP ou Telnet) à un client RADIUS (ou NAS : *Network Access Server*), qui est en général une passerelle/proxy.
2. Le client RADIUS demande à l'utilisateur son nom et son mot de passe, et il les communique de manière sécurisée à un serveur RADIUS relié à une base de données ou à un annuaire LDAP.
3. En fonction de la zone d'accès demandée et des droits de l'utilisateur, le serveur RADIUS peut exiger des informations supplémentaires pour l'authentification.

- La réponse CHALLENGE permet d'éviter de transmettre le mot de passe. Le client envoie alors une autre requête répondant au Challenge pour s'authentifier.
  - Si l'identification réussie, le Serveur répond par un ACCEPT (REJECT dans le cas contraire).
4. Le serveur RADIUS envoie enfin les autorisations de l'utilisateur. Il pourra par la suite bloquer une connexion en cours et assurer une journalisation des accès.

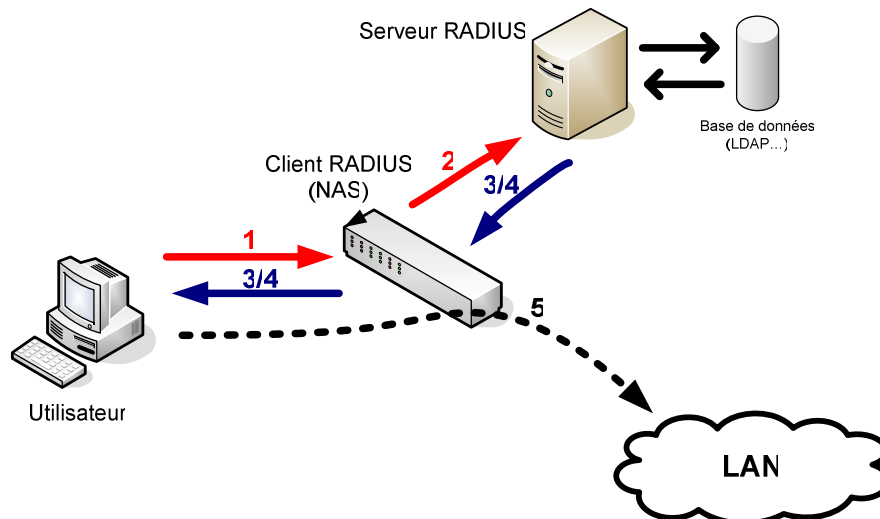


Fig. 12 : Fonctionnement de Radius

#### Les limitations du protocole :

- Il a été conçu au départ pour des identifications sur des liaisons lentes et peu sûres. Le choix du protocole UDP (port 1812) conduit à des échanges laborieux basés sur des temporisations de réémission et des échanges d'accusé de réception.
- Sécurité relative reposant sur le secret partagé. Certaines implémentations clientes limitent en plus sa taille.
- Chiffrement de l'attribut User-password par une fonction de hashage MD5, plutôt réservé pour des opérations de signature.
- Le rejeu des réponses du serveur est possible.
- Pas de mécanisme d'identification du serveur. Il est ainsi possible de se faire passer pour un serveur RADIUS et de récolter les noms et mots de passe des utilisateurs.

Les normes qui complètent le protocole RADIUS sont les protocoles d'authentification PAP, CHAP ou EAP. Le successeur du protocole Radius pourrait être le protocole Diameter. Nous en parlerons plus loin.

Les différentes implémentations de RADIUS sont :

- Microsoft : le service d'authentification IAS pour Windows Serveur 2000/2003, et NPS (Network Policy Server) pour Windows Server Vista.
- Communauté du libre : OpenRadius et FreeRadius.

#### IAS et NPS

IAS (*Internet Authentication Service*) est le service d'authentification Internet sur Windows 2000 (Serveur IAS) et Windows Server 2003 (Service IAS). C'est une implémentation Microsoft du serveur RADIUS : Le service IAS joue le rôle du serveur RADIUS. Il effectue une authentification, une autorisation et une gestion des comptes centralisées des connexions pour de

nombreux types d'accès réseau (accès sans fil, accès par commutateur d'authentification, accès par connexion à distance et VPN). En tant que proxy RADIUS, le service IAS peut envoyer les messages d'authentification et de gestion de comptes à d'autres serveurs RADIUS.

NPS (*Network Policy Server*) est l'implémentation d'un serveur et proxy RADIUS sur Windows Serveur Vista. Il remplace le service d'authentification Internet (IAS) de Windows Server 2003. Cette nouvelle implémentation effectue toutes les fonctions déjà présentes dans IAS notamment pour les connexions VPN, pour les connexions sans fil et pour les connexions basées sur 802.1X. De plus, NPS effectue une évaluation du bon fonctionnement du réseau et réserve un accès limité ou illimité pour les clients NAP.

NAP (*Network Access Protection*) est une nouvelle protection de l'accès réseau dans Windows Server Vista. Il offre de nouvelles possibilités au niveau stratégie de sécurité : Par exemple il peut demander que les entités du réseau possèdent les dernières mises à jour du système d'exploitation et les derniers fichiers de signature antivirus. En fonction de cela, les entités auront plus ou moins de droits sur le réseau. Ces entités sont appelés client NAP.

NPS prend en charge aussi l'envoi du trafic RADIUS via IPv6 (RFC 3162).

## 5.2- TACACS+

Tout comme Radius, TACACS+ (*Terminal Access Controller Access Control System Plus*) est un serveur d'authentification permettant de centraliser les autorisations d'accès dans un parc d'entreprise. Ce protocole inventé par CISCO Systems a remplacé TACACS et XTACACS mais n'est pas basé sur ceux-ci. TACACS+ supporte différents types d'architectures, par exemple, si un utilisateur utilise une connexion point à point, c'est le serveur d'accès qui va jouer le rôle de client TACACS+ pour interagir avec le serveur TACACS+, en revanche, si l'utilisateur utilise une connexion Wifi, c'est le point d'accès qui va jouer le rôle de client TACACS+. D'autre part, TACACS peut être utilisé pour s'authentifier sur un matériel CISCO, dans ce cas la connexion est initiée sur le matériel CISCO, c'est ce dernier qui va interroger le serveur RADIUS.

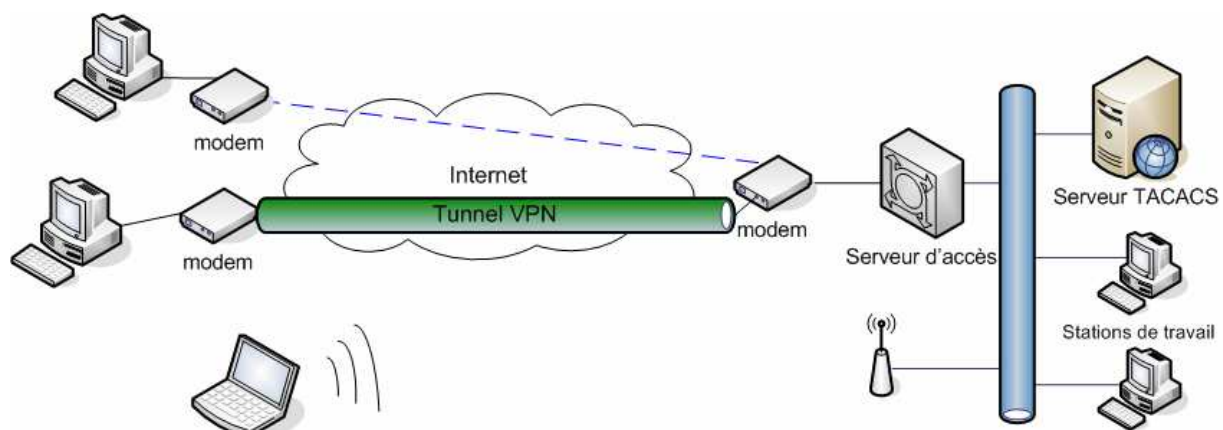


Fig. 13 : Différentes architectures supportées par TACACS

La particularité de TACACS+ par rapport aux serveurs d'authentification traditionnels est la séparation protocolaire des trois fonctions AAA (*Authentication, Authorization, Accounting*). En effet, TACACS+ permet d'utiliser des technologies différentes, que ce soit pour déterminer l'identité d'une personne, de déterminer ses droits, ou encore de gérer l'enregistrement des logs. TACACS utilise le protocole TCP (port 49) et généralement une seule session (authentification, autorisation ou enregistrement de logs) est réalisée par connexion TCP. Chaque session est numérotée et les paquets peuvent être chiffrés par cet identifiant de sessions.

La phase d'authentification peut supporter plusieurs protocoles comme des techniques de type PAP (login - mot de passe) ou encore CHAP (à base de challenge). Ensuite, quelque soit la fonction AAA demandée au serveur TACACS+, une session suit toujours le même protocole : le client envoie une requête START au serveur décrivant le type de session initiée, puis, des paires de messages de type REQUEST/REPOONSE contenant des paires « attributs-valeur ». La RFC ne définit pas d'implémentation spécifique pour le stockage des informations relatives aux comptes utilisateur, le serveur TACACS+ peut aussi bien utiliser des fichiers systèmes (/etc/passwd), des bases de données, des cartes à puces ou encore d'autres serveur d'authentification comme Kerberos.

TACACS+ est donc un serveur d'authentification relativement simple mais il couvre quand même l'ensemble des fonctions AAA et de plus il peut s'intégrer à tout type d'infrastructure de part sa liberté d'implémentation.

### 5.3- Kerberos

Kerberos est un protocole d'authentification réseau standardisé par l'IETF. L'objectif de Kerberos est double : sécuriser un échange sur un réseau non sécurisé et avoir une authentification fiable de l'utilisateur. Il est basé sur deux entités :

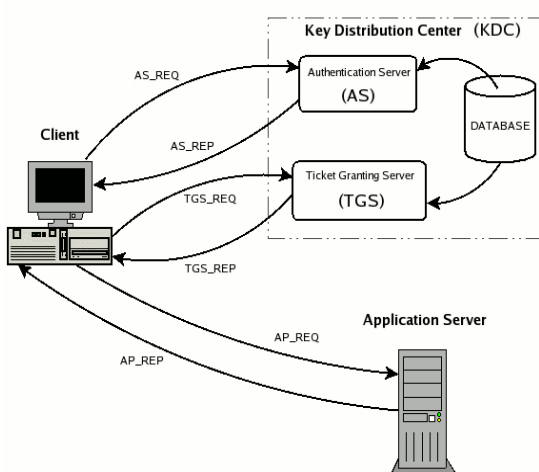
- Le serveur d'authentification (AS : *Authentication Server*) qui prend en charge toute la partie authentification pur du client. C'est lui seul qui peut permettre au client de communiquer au TGS (grâce à un ticket d'accès).
- Le serveur de distribution de tickets (TGS : *Ticket Granting Server*) prend en charge les demandes d'accès aux services des clients déjà authentifiés.

L'ensemble des infrastructures serveur de Kerberos AS et TGS est appelé le centre de distribution de clés (KDC : *Key Distribution Center*). Ils sont généralement regroupés sur le même serveur.

Dans Kerberos, tous les tiers doivent prouver leur identité : on utilise des mécanismes d'authentification mutuelle. Le protocole est basé sur des tickets horodatés et chiffrés. Les échanges reposent sur un système de cryptographie (algorithme DES) à base de clés symétriques. Kerberos partage avec chaque client du réseau une clé secrète faisant office de preuve d'identité.

#### Les mécanismes d'authentification :

Le client désire accéder au serveur pour obtenir un service. Ce serveur est représenté dans notre schéma par l'*Application Server*.



- Le client a sa propre clé privée  $K_c$ .
  - Le serveur a sa propre clé privée  $K_s$ .
  - Le TGS a sa propre clé privée  $K_{tgs}$  et connaît la clé privée du serveur  $K_s$
  - L'AS connaît les clés privées du client et de TGS.
- Le TGS et AS sont deux entités normalement de confiance.

1. AS\_REQ : le client s'identifie auprès de l'AS à l'aide d'un mot de passe ou d'une carte à puce.
2. L'AS vérifie dans sa base (par exemple AD) que le client existe. Il génère une clé de session  $K_{c,tgs}$ .

Puis il envoie au client AS\_REP :

- Une clé de session  $K_{c,tgs}$  chiffrée avec  $K_c$ , qui fera office de mot de passe temporaire pour chiffrer les communications suivantes.
  - Un ticket d'accès T1 au service de délivrement de ticket, chiffré avec  $K_{tgs}$  (que le client ne peut donc pas déchiffrer). Il contient notamment l'heure de l'opération, sa durée de validité, l'adresse de la machine cliente ainsi que la clé de session  $K_{c,tgs}$ .
3. TGS\_REQ : le client fait une demande de ticket auprès du TGS. Le client lui transmet :
- Le ticket d'accès T1 que l'AS lui avait donné .
  - Un identifiant contenant des informations sur le client avec la date d'émission, chiffrées avec la clé de session  $K_{c,tgs}$
4. Le TGS :
- Déchiffre avec sa clé, le ticket d'accès T1. Il obtient la clé de session  $K_{c,tgs}$ . Le TGS est maintenant certain que le client a bien obtenu le T1 de l'AS.
  - Déchiffre alors les informations que le client avait précédemment chiffré avec la clé de session.
- Il vérifie que la durée de validité est correcte. Puis le TGS envoie au client TGS\_REP qui comprend :
- Un ticket T2 pour accéder au serveur d'application. Il est chiffré avec la clé privée de ce serveur  $K_s$ .
  - Une seconde clé de session  $K_{c,s}$  pour les communications entre le serveur final et le client. Cette clé a été chiffrée avec la clé initiale  $K_{c,tgs}$ .
5. Le client déchiffre la seconde clé de session  $K_{c,s}$  avec  $K_{c,tgs}$
- Il envoie AP\_REQ au serveur d'application deux informations :
- Un nouvel identifiant chiffré avec  $K_{c,s}$
  - Le ticket d'accès T2
6. Le serveur d'application vérifie que le ticket est valide en déchiffrant T2 avec  $K_s$ . Il obtient  $K_{c,s}$ .

Le serveur peut alors vérifier la cohérence entre les deux informations. Par exemple il vérifie que la demande est conforme à ce qui est autorisé par le ticket. Une réponse positive ou négative AP\_REP est envoyée au client.

#### **Les points forts de Kerberos :**

- Le transit des mots de passe sur le réseau est chiffré.
- Il permet aux utilisateurs de s'authentifier une fois pour toutes lors du login. Ils pourront après utiliser tous les services d'accès à distance sans avoir à fournir à chaque fois leur login et mot de passe. Ils sont en fait toujours authentifiés de manière transparente par Kerberos pour eux.
- Séparation des rôles : l'AS et le TGT. C'est la base de Kerberos. Mais dans la réalité, ces deux rôles sont regroupés en une même entité (KDC).
- Impossible de rejouer un échange deux fois de la même manière (grâce au timestamps).

#### **Les points faibles de Kerberos :**

- Le chiffrement symétrique nécessite un partage des clés entre l'AS et le client.
- Les horloges doivent être parfaitement synchronisées : en effet, l'anti-rejeu s'appuie sur le « timestamps ».
- L'authentification mutuelle n'est pas disponible lors du premier échange entre l'AS et le client. Le client ne peut pas certifier que l'AS est bien celui qu'il prétend être.

En revanche, le client peut exiger que le serveur d'application s'authentifie à son tour (lors de la dernière étape). Ce dernier s'exécute en renvoyant la date courante (plus récente que celle du précédent message du client) chiffrée avec  $K_{c,s}$ . Etant donné que seuls le client et le serveur connaissent  $K_{c,s}$ , le client peut raisonnablement penser que c'est bien le serveur qui lui répond.

Remarque : Kerberos est le mécanisme d'authentification par défaut dans Windows pour vérifier l'identité d'un utilisateur ou d'un ordinateur. Les rôles de l'AS et TGS sont pris en compte par le contrôleur de domaine, en s'appuyant sur l'annuaire Active Directory.

On pourrait distinguer deux types d'authentifications sous Windows :

- Il y a l'ouverture de session interactive qui permet de s'authentifier en local sur l'ordinateur. Elle s'appuie sur le référentiel utilisateur local appelé base SAM (*Security Accounts Manager*)
- Il y a l'authentification réseau qui permet d'accéder à un ordinateur distant pour en parcourir les répertoires. Elle s'appuie sur la base SAM de l'ordinateur distant.

Il existe aussi un mécanisme (notion de domaines Windows) qui permet de mixer les deux : Ainsi il est possible de s'authentifier sur l'ordinateur local en utilisant une base de comptes utilisateurs stockés sur un serveur réseau. Pour gérer les données du réseau, Windows utilise un annuaire : Active Directory.

## 5.4- Authentification web

Comme nous l'avons montré ci-avant, les protocoles SSL et TLS permettent de mettre en place un service d'authentification (et de chiffrement) pour http, mais même s'il est possible d'effectuer une authentification du client grâce à TLS, l'objectif principal de ces protocoles est de permettre d'être certain de l'authenticité du serveur. Pour mettre en place une authentification des clients il existe plusieurs méthodes applicatives.

### 5.4.1- PHP/ASP

La méthode la plus simple pour réaliser une authentification des utilisateurs avec un langage scripté tel que PHP ou ASP est de créer un formulaire permettant de saisir login et mot de passe puis de créer un page PHP ou ASP appelée par le formulaire et vérifiant que le login et le mot de passe sont valide en les comparant à ceux stockés en dur dans la page PHP ou ASP. La page d'authentification étant exécutée côté serveur il est à priori impossible pour un client de lire son contenu.

La page PHP suivante permet de réaliser ce type d'authentification :

```
<?php      if ($_POST['pseudo'] = login_valide && $_POST['passwd'] = passwd_valide)
            header('Location: authorized_page.php');
            else
            echo 'Pseudo ou mot de passe erroné !';
?>
```

Les inconvénients de ce type d'authentification sont bien évidemment nombreux : d'une part n'importe qui ayant accès à la page depuis le serveur peut lire les login et mots de pass, et d'autre part la maintenance des comptes est fastidieuse et non centralisée.

La méthode la plus utilisée est donc généralement un interfaçage avec une base de données. PHP permet par défaut une connexion avec un très grand nombre de bases de données, et si une base n'est pas supportée ou alors si ASP est utilisé, le pilote propriétaire ODBC (*Open DataBase Connectivity*) développé par Microsoft peut être utilisé. Pour réaliser l'interfaçage, la page PHP/ASP doit définir les propriétés de la base de données : nom d'utilisateur, mot de passe, adresse réseau de la base de données, et nom de la base afin de pouvoir initier une connexion avec elle. Puis, une fois la connexion effectuée, il suffit d'utiliser des requêtes SQL du type : *"SELECT login, passwd FROM authorized\_accounts"* afin de vérifier les permissions des utilisateurs.

### 5.4.2- .htaccess

Les fichiers «.htaccess » sont des fichiers texte propres au serveur web Apache. Ils permettent de définir des règles d'accès à toute une partie, voir tout le site hébergé. Ils sont stockés dans le répertoire sur lequel on désire effectuer l'authentification et contiennent une instruction par ligne : Tout d'abord il faut spécifier l'endroit où sont contenus les login/mot de passe des visiteurs autorisés (AuthUserFile, AuthGroupFile) puis il faut définir le message de la boîte de

dialogue qui va s'afficher pour que l'utilisateur rentre son login et son mot de passe. Enfin, entre les balises <limit GET POST> ou <limit PUT POST> (suivant le type de requête) et </limit> il faut spécifier quels sont les utilisateurs autorisés (par exemple « *require user Toto* »). Les fichiers « .htaccess » permettent également de filtrer les utilisateurs par IP ou d'utiliser MD5 pour réaliser un hash des mots de passe.

## 6- Annuaire

### 6.1- Active Directory

Active Directory est un annuaire système hiérarchique et compatible LDAP. Il permet de localiser, rechercher et gérer des ressources représentées par des objets de l'annuaire. Il offre des mécanismes de sécurité pour protéger ses informations. Il permet de gérer des ressources liées à la gestion du réseau (domaines, comptes utilisateurs, stratégies de sécurité etc...). La base de données d'AD est distribuée ce qui lui améliore la tolérance aux pannes.

Active Directory respecte le standard LDAPv3. Il est donc capable d'interagir avec des clients et des serveurs LDAP d'autres origines. Les protocoles d'échange entre serveurs AD sont propriétaires et non publics. Un contrôleur de domaine ne pourra donc pas travailler avec l'annuaire d'un fournisseur tiers. Certains produits Microsoft sont installés par défaut (ou fortement conseillés lors de l'installation): DNS, serveur WeB. D'autres bénéficient d'une forte intégration avec AD (serveur de courrier Exchange, ISA Server).

Active Directory centralise l'authentification. Le contrôle d'accès peut être défini à la fois sur chaque objet de l'annuaire et sur chaque propriété de ces objets. Il fournit non seulement le stockage mais également l'étendue d'application des stratégies de sécurité.

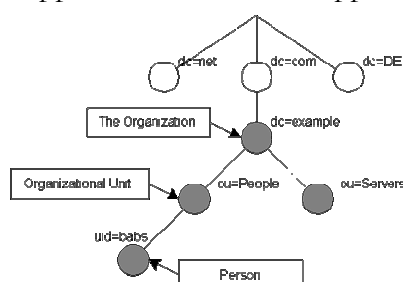
Quels que soient les qualités des produits concurrents (Serveurs Apaches, annuaires open-ldap ou novell, serveurs DNS etc...), leur mise en place sera forcément moins naturelle que celles des produits Microsoft. AD est l'un des maillons de la conquête du marché des serveurs par Microsoft. Le support par AD d'un certain nombre de protocoles standard a pour but de fédérer l'ensemble des ressources réseau autour de serveurs Microsoft.

Nous ne rentrerons pas dans les détails techniques de configuration d'AD. Mais nous allons plutôt nous attaché aux mécanismes généraux qu'il utilise : LDAP et KERBEROS notamment.

### 6.2- LDAP

LDAP (*Lightweight Directory Access Protocol*) est actuellement en version 3 et il est normalisé par l'IETF. Il s'agit d'un protocole d'interrogation d'annuaire. On dit qu'il est allégé, par comparaison à la norme X500, son ancêtre, dont la mise en œuvre était très lourde.

LDAP au cœur du système d'information. Il regroupe les données d'une entité au même endroit. On dit que c'est un annuaire fédérateur. C'est un standard incontournable : la plupart des applications récentes s'appuient dessus : les outils de messageries, les actifs du réseau (proxy, firewall...), les progiciels de gestion, les intranets etc... Une majorité de logiciels utilisent LDAP pour l'authentification.



LDAP est une base donnée hiérarchique et non pas relationnelle comme les SGBD. On peut ainsi représenter son contenu sous forme d'arbre. L'annuaire est conçu pour

référencer toutes sortes d'informations : informations sur des personnes, sur des applications, sur un parc informatique...

Le concept de l'annuaire, c'est de maintenir de façon cohérente et contrôlée une grande quantité de données et d'optimiser la consultation en lecture au dépend de l'écriture. Les accès concurrents sont gérés et la quantité de donnée pouvant être stockée est quasi illimitée. La contrepartie est que les mises doivent être ponctuelles et les enregistrements peu volumineux : On utilise des chaînes de caractère courtes et on limite le nombre de fichiers binaires (certificats, photos...).

LDAP propose donc des mécanismes pour gérer l'authentification. Plusieurs méthodes sont possibles en fonction du niveau de sécurité désiré :

- La connexion anonyme est généralement limitée à la consultation de parties restreintes de l'annuaire.
- L'authentification par login/mot de passe.
- L'authentification par login/mot de passe avec hachage de ce dernier.
- L'authentification par login/mot de passe sur TLS avec un tunnel TLS entre le client et l'application et un tunnel TLS entre l'application et l'annuaire.
- L'authentification par certificat X509.
- Authentification plus élaborée grâce aux API SASL (*Simple Authentication and Security Layer*). Cette API permet d'intégrer facilement des mécanismes d'authentification forte comme Kerberos, Radius, ou des systèmes de mot de passe à usage unique comme OTP.

Il est aussi possible de sécuriser les échanges entre l'annuaire LDAP et les clients par LDAP sur TLS ou LDAPS.

Le principal avantage de LDAP est la normalisation de l'authentification. Il est très facile de programmer un module d'authentification reposant sur LDAP à partir d'un langage possédant une API LDAP. C'est l'opération Bind qui permet d'authentifier un utilisateur. Aujourd'hui, de plus en plus d'applications Web possèdent un module d'authentification prenant en charge LDAP.

Les applications peuvent ainsi déléguer l'étape d'authentification à l'annuaire LDAP qui met en œuvre l'une des méthodes précédentes. Par exemple sur les systèmes GNU/Linux récents, les fichiers password et shadow sont de plus en plus remplacés par des appels à LDAP. Les données peuvent être accédées par le module PAM (*Pluggable Authentication Modules*).

PAM est un mécanisme (sous Linux, Solaris, HP-UX) qui permet justement de déléguer les fonctionnalités d'authentification, sans recompiler quoi que ce soit. PAM peut être paramétré en fonction de l'application (base de compte NTLM pour les anciennes versions de Windows, base de compte Unix ou autre) et en fonction du mode d'accès à l'application (Intranet, Extranet, Web). Le module de connexion de PAM fournit l'interface avec les annuaires (ou les fichiers). C'est cette couche qui validera que l'identifiant et le mot de passe sont valides et retournera une réponse aux applications.

### 6.3- NIS/NIS+

NIS (*Network Information Service*) est un système de base de données distribuée fonctionnant au dessus de RPC et permettant de gérer et stocker les informations systèmes (les noms d'hôte /etc/hosts, les comptes utilisateurs /etc/passwd, etc...) et de les diffuser à tous les hôtes du réseau. Originellement, NIS portait le nom Yellow Pages (YP) mais Sun l'a renommé NIS, cependant, toutes les commandes NIS commencent par yp.

NIS est architecturé en différents domaines, chaque domaine contient : un serveur NIS maître qui maintient la base de données, des serveurs esclaves optionnels qui déchargent le serveur maître et font office de sauvegarde, et des clients NIS qui peuvent interroger les serveurs

maîtres ou esclaves. Le serveur maître est le seul à pouvoir modifier la base de données (changements de mots passe, etc.). Dès qu'une modification est effectuée sur la base, elle est automatiquement propagée aux serveurs esclaves.

Le principal inconvénient de NIS est son manque de sécurité : n'importe quel hôte du réseau peut interroger le serveur sans avoir à s'authentifier. Une évolution de NIS plus sécurisée et permettant de gérer de gros réseaux a été développée par Sun : NIS+. Cette version totalement reconçue supporte le chiffrement des données et l'authentification sécurisée par RPC. Malgré ces améliorations, NIS et NIS+ ne font plus l'objet de développements spécifiques et sont remplacés par LDAP, beaucoup plus puissant et complet.

## 7- Les évolutions à venir

### 7.1- Diameter

Diameter, une solution d'authentification utilisant le protocole TCP au niveau de la couche transport, au lieu d'UDP dans le cas de Radius.

Il est de toute manière compatible avec les normes Radius et EAP. Ce n'est pas une extension de RADIUS mais un successeur du protocole. Il est basé sur TCP et utilise des attributs de grande taille. Diameter est destiné aux échanges entre serveurs sur des liaisons sûres ; les serveurs Diameter (non déployés pour l'instant) sont généralement compatibles Radius. Un certain nombre de types d'attributs Diameter se retrouvent déjà dans EAP-TTLS par exemple.

### 7.2- Liberty Alliance

Créé par un consortium d'éditeur (notamment Intel, France-Telecom, Verisign, IBM, Sun...) afin de contrer le système propriétaire Passport de Microsoft. Passport est en effet un moteur SSO (*Single Sign On*) qui consiste à simplement s'authentifier une fois sur le système d'information. Les sessions sont alors propagées aux applications s'appuyant sur ce système d'authentification unique.

Liberty propose de standardiser l'invocation des services d'identité en définissant un protocole basé sur les standards et indépendant des langages de programmation. Il définit une infrastructure basée sur des « fournisseurs de services » et des « fournisseurs d'identités » :

- Les fournisseurs de services sont les applications nécessitant une authentification, donc consommatrices de données utilisateur.
- Les fournisseurs d'identité délivrent ces informations utilisateurs.
- Ces acteurs sont reliés entre eux par des cercles de confiance définissant des accords de coopération. Ils s'échangent des informations sur l'identité de leurs utilisateurs respectifs. Elles présument qu'en amont lesdits utilisateurs ont été correctement authentifiés.

La grande innovation des fournisseurs d'identité est d'offrir, en plus de l'authentification, une série de services autour du référentiel utilisateur :

- Un service de SSO (*Single Sign On*) qui permet de propager des sessions utilisateurs entre fournisseurs de services.
- Un service de *Global Logout*, qui permet de mettre fin à l'ensemble des sessions chez les fournisseurs de services.
- Un service de « profil utilisateurs » qui permet de séparer les données « profils utilisateurs » et d'en réglementer l'accès.

Le projet Liberty décrit une série de services d'identité « user-centric » grâce auxquels, leur usager peut gérer l'accès à ses données personnelles. Liberty Alliance apporte aussi la notion de

fédération. Au lieu que ce soit un fournisseur de service qui décide si un utilisateur a le droit d'accéder à son service sans se ré-identifier, c'est l'utilisateur qui décide s'il veut accéder à ce service sans se ré-identifier.

D'autres nouveaux concepts sont apportés : comme une nouvelle technologie de chiffrement permettant à l'utilisateur de masquer ses données personnelles au fournisseur d'identité central, ainsi qu'aux fournisseurs de services dans chaque "cercle de confiance". L'utilisateur peut garder ainsi la mainmise sur les informations qu'il accepte de voir transiter entre deux sites de commerce en ligne. Reste ensuite à trouver un point d'équilibre entre, d'un côté, la tendance à enfermer le client dans un système où il est tenu à faire confiance au fournisseur d'identité, et de l'autre, la valeur ajoutée que constitue la complète maîtrise de ses données privées pour l'utilisateur.

Enfin, Liberty Alliance n'a pas confiance dans les mots de passe. L'authentification forte s'impose petit à petit. C'est une authentification forte à base de jetons, cartes à puce, biométries ou produits similaires. Les mots de passe sont jugés dépassés pour une sécurité digne de ce nom. Elle évite le risque d'usurpation d'identités, le seul mot de passe ne suffisant plus à accéder aux services.

Si ce standard est en cours de finalisation (une nouvelle version est sortie en novembre 2006), il commence à être implémenté par les grands éditeurs (Sun, Novell, SAP etc...).

### 7.3- La biométrie

La biométrie apparaît comme le meilleur moyen de s'authentifier, de façon fiable et pratique. C'est difficilement falsifiable. On distingue les moyens de biométrie physiques (la main, les yeux), des moyens de biométrie issus du comportement de la personne (la voix, la signature).

Il y a différents moyens biométriques pour s'authentifier :

1. La géométrie de la main : Identifier quelqu'un par la forme de sa main et/ou ses empreintes digitales sont des moyens faciles à utiliser et qui offre des bonnes performances.
2. La rétine : Identifier une personne en analysant son empreinte « rétinienne », c'est-à-dire la disposition des vaisseaux sanguins du fond de l'œil. Pour récupérer cette empreinte, il faut obliger l'utilisateur à regarder dans un dispositif qui éclaire le fond de l'œil. En général, les gens sont assez réticents à ce moyen d'authentification. Pourtant, ce serait le moyen le plus sûr.
3. L'iris : Son motif est unique pour chaque individu. L'authentification via l'iris ne nécessite pas d'éclairer la rétine.
4. Le visage : Identifier une personne sur les caractéristiques de son visage : distance entre le nez et la bouche, entre les yeux...
5. La signature : Identifier quelqu'un sur la façon dont il signe un texte : la vitesse, la pression exercée ...
6. La voix : Identifier une personne sur la tonalité de sa voix (caractéristiques de ses fréquences). Ce n'est pas sur la reconnaissance de ce qu'elle dit.

Caractéristiques	Empreinte digitale	Géométrie de la main	Rétine	Iris	Visage	Signature	Voix
Facilité d'utilisation	☼☼☼☼	☼☼☼☼	☼	☼☼	☼☼	☼☼☼☼	☼☼☼☼
Cause d'erreur	Moiteur, la saleté, âge.	Blessure à la main, l'âge.	Lunettes	Éclairage trop faible.	Éclairage, l'âge, les lunettes, les	Changement dans la signature.	le bruit, le temps, un rhume

					cheveux.		
Caractéristiques	Empreinte digitale	Géométrie de la main	Rétine	Iris	Visage	Signature	Voix
La précision	☼☼☼☼	☼☼☼☼	☼☼☼☼☼	☼☼☼☼☼	☼☼☼☼	☼☼☼☼	☼☼☼☼
Acceptation du public	☼☼☼☼	☼☼☼	☼☼☼	☼☼☼	☼☼☼	☼☼☼	☼☼☼☼
Niveau de sécurité requis	☼☼☼☼	☼☼☼	☼☼☼☼☼	☼☼☼☼☼	☼☼☼	☼☼☼	☼☼☼
Stabilité dans le temps	☼☼☼☼	☼☼☼	☼☼☼☼☼	☼☼☼☼	☼☼☼	☼☼☼	☼☼☼

Tableau trouvé sur <http://www.securite.teamlog.com>

Pour mettre en place un système d'authentification biométrique, il faut :

- Dans un premier temps : récupérer les données et les stocker dans un endroit sécurisé.
- Ensuite, lors de l'authentification : Il faut récupérer les données stockées, récupérer la nouvelle « empreinte » puis comparer les deux "empreintes" biométriques.

La mise en place d'un moyen de biométrie implique du matériel pour récupérer l'information (le capteur), une base de données pour récupérer les informations, une recherche pour déterminer au mieux les besoins. Elle implique aussi la mise en place et l'installation et la connexion du système côté utilisateur, la formation des utilisateurs et la mise en place de procédures d'exception quand une personne n'arrive pas à s'identifier, sans oublier la maintenance du système.

Au final, cela demande une certaine adaptation au mécanisme au début.

On peut aussi imaginer l'utilisation de moyens biométriques dans une PKI (Public Key Infrastructure), où les utilisateurs possèderaient une carte à puce avec identification d'une empreinte digitale en plus. La biométrie est une solution d'authentification forte qui gagne petit à petit du chemin.

## 8- Conclusion

Notre étude montre que le système d'authentification parfait n'existe pas, aucune méthode n'est générique et il est indispensable de connaître parfaitement l'infrastructure réseau et système pour choisir le système d'authentification le plus adapté. Il faut également bien distinguer les méthodes d'authentification faibles des méthodes d'authentification forte : il est certain qu'à l'heure actuelle les systèmes basés sur des mots de passe sont moins robustes que les systèmes à base des certificats.

Cependant quelles que soient les données sur lesquelles se base une authentification, si elles sont numériques, il y a toujours un risque de vol d'identité. C'est pour cette raison que l'avenir se tourne donc plutôt vers des techniques à priori infalsifiables comme la biométrie. Le futur s'oriente également vers des systèmes d'identité centralisés pouvant être propagés. Ces systèmes d'authentications uniques permettent non seulement de simplifier l'accès aux services protégés mais surtout de limiter les risques potentiels liés aux authentifications multiples.

Enfin, il ne faut pas oublier que la sécurité est un tout, et que l'authentification en est la première brique, elle doit impérativement être complétée, par exemple par des technologies de gestion des droits ou de chiffrement.

## 9- Bibliographie

### 9.1- Documentations écrites

- G. Plouin, J. Soyer et M-E. Trioullier : *Sécurité des architectures Web*, Dunod, 2004.  
 G. Pujolle : *Initiation aux Réseaux*, Eyrolles, 2000.  
 A. Tanenbaum : *Réseaux 4<sup>e</sup> Ed*, Pearson Education, 2004.  
 S. Borderes, N. Makarevitch : *Authentification réseau avec Radius*, Eyrolles, 2006.  
 J. Garman : *Kerberos: The Definitive Guide*, O'reilly, 2003.  
 J. Bournelle et M. Laurent-Maknavicius : *Adaptation et implémentation de Diameter/AAA pour Mobile IPv6*, Proceeding of the DNAC 2002, 2002.  
 Y. Legrandgérard : *Le protocole IPSec et les Réseaux Virtuels Privés*, 2006

### 9.2- Documentations en lignes

#### Protocoles d'authentification :

- <http://www.labo-cisco.com/ArticleComp.asp?ARID=30>  
<http://www.ietf.org/rfc/rfc2284.txt>  
<http://open1x.sourceforge.net/>

#### NuFW :

- <http://www.nufw.org>

#### SSL/TLS :

- <http://www.certa.ssi.gouv.fr/site/CERTA-2005-REC-001>  
<http://www.openssl.org>  
<http://www.ietf.org/rfc/rfc2246.txt>

#### RADIUS :

- <http://www.freeradius.org>  
<http://www.ietf.org/rfc/rfc2865.txt>

#### TACACS+ :

- <http://www.ietf.org/rfc/rfc1492.txt>  
<http://www.cisco.com/warp/public/614/7.html>

#### Kerberos :

- <http://www.ietf.org/rfc/rfc4120.txt>

#### Authentification web :

- <http://www.securiteinfo.com/conseils/htaccess.shtml>

#### Active Directory

- <http://www.lami.univ-evry.fr/~petit/>

#### LDAP :

- <http://www.ietf.org/rfc/rfc4403.txt>  
<http://www.cru.fr/ldap/>

#### NIS/NIS+ :

- <http://tldp.org/HOWTO/NIS-HOWTO/>

#### Liberty Alliance :

- <http://www.projectliberty.org/>

#### Autres :

- <http://www.securite.teamlog.com>  
<http://www.wikipedia.org>  
<http://www.Microsoft.com>  
<http://www.commentcamarche.net>  
<http://www.supinfo-projects.com>  
<http://www710.univ-lyon1.fr/~ogluck>  
<http://graal.ens-lyon.fr/~ycaniou>  
<http://www.securite.org/>  
<http://www.ietf.org>