REPUBLIQIUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du

Diplôme d'Ingénieur d'Etat en Electronique

Thème :

Déploiement d'un réseau sécurisé sans fil Wifi pour un campus universitaire sous un fournisseur d'identité Shibboleth

Proposé et dirigé par :

Mr. R. SADOUN

Réalisé par :

Mr. Mohamed Yassine GACEM
Mr. Abdelrezak OUGHLIS

Soutenu le Samedi 27 Juin 2009 devant le jury composé de :

Président : *Pr. A. BELOUCHRANI*
Rapporteur: *Mr. R. SADOUN*
Examinateur: *Pr. M. MEHENNI*
Examinateur: *Mr. M. KHODJA*

-Promotion: Juin 2009-

Remerciements :

Tout d'abord nous remercions le Bon DIEU, le Clément de nous avoir donné la foi et le courage de terminer ce projet.

Nos profonds remerciements vont à notre promoteur Mr R.SADOUN qui nous a donné l'occasion de travailler sur un sujet passionnant, pour sa confiance, ses conseils judicieux et sa collaboration.

Nous remercions Pr A.BELOUCHRANI, enseignant à l'Ecole Nationale Polytechnique, de nous avoir fait l'honneur de présider le jury et pour ses encouragements de valeur.

Nos remerciements vont aussi aux Pr M.MEHENNI et Mr M.KHODJA pour avoir accepté d'examiner ce modeste travail.

Nos profondes gratitudes à toutes les personnes ayant contribué à notre formation.

Mohamed Yassine GACEM

Abdelrezak OUGHLIS

Dédicace :

Je dédie ce projet de fin d'études, aux personnes qui me sont les plus chères :

A mes parents ***Halima*** *et* ***Mustapha*** *qui m'ont énormément soutenu dans les moments les plus difficiles, partagé mes joies et mes peines, qui se sont toujours sacrifiés pour moi.*

A mon frère aîné ***Youcef*** *pour son soutien continu pendant toute ma formation.*

A mon frère ***Mehdi*** *pour ses encouragements.*

A mes grands parents ***Mama Hadja***, ***Djedi*** *et* ***Mamie*** *pour leurs amours.*

A toute ma famille.

A mes amis et collègues.

Mohamed Yassine GACEM

Dédicace :

Je dédie ce modeste travail à :

A ma très ***chère mère*** *qui a veillé sur moi pendant toute ma vie et*

Mon ***cher père*** *défunt que Dieu accueille son âme dans son vaste paradis.*

Mon ***frère*** *et ma* ***sœur****.*

Ma très chère famille.

Tous les profs qui m'ont soutenu de l'école nationale polytechnique.
Tous mes amis.

Abdelrezak OUGHLIS

Sommaire

Page

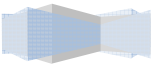
Chapitre 4 : Développement et Application

Liste des Figures

Page

Liste des Tableaux

Page



Introduction Générale

L'accès à l'information et l'échange des données entre utilisateurs ont connu un développement époustouflant. Les réseaux sans fil ont emmené un confort et une facilité indéniables à cet accès et à cet échange

Les réseaux sans fil ont solutionné beaucoup de problèmes que connaissaient les réseaux filaires, à savoir la difficulté de mise en place, la mobilité quasi difficile ainsi que le coût.

Sachant que chaque système informatique a plusieurs failles exposant son utilisation à des risques qui peuvent être parfois très dangereux, un réseau sans fil est sans aucun doute un appât pour les intrus.

Ce rapport a pour but de présenter une solution de déploiement et de sécurisation (par le biais d'une fédération d'identité) d'un réseau sans fil WiFi au sein d'un campus universitaire.

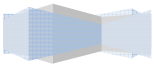
Ce rapport est composé de quatre parties chacune centrée sur une ou plusieurs briques :

- Généralités sur le WiFi et l'intégration de la sécurité ;
- Fédération d'identité ;
- Le Portail Captif ;
- Le Développement et le déploiement de la solution;

La première partie a pour but d'apporter une vue d'ensemble pour nous permettre de bien préparer et réaliser le déploiement de notre réseau sans fil, dans le chapitre 2, la partie sécurité sera abordée.

Par la suite, le chapitre 3 exposera Shibboleth et détaillera les outils nécessaires pour déployer un SSO sous une fédération d'identité afin de sécuriser le réseau au niveau applicatif.

Le chapitre 4 exposera les différentes étapes suivies pour réaliser ce projet. En dernier, nous terminerons par une conclusion.



Chapitre 1 :

Généralités sur le réseau WiFi

1.1. Introduction :

La norme *IEEE 802.11* (*ISO/IEC 8802-11*) est un standard international décrivant les caractéristiques d'un réseau local sans fil (*WLAN*). Le nom **WiFi** (contraction de *Wireless Fidelity*, parfois notée *Wi-Fi*) correspond initialement au nom donné à la certification délivrée par la Wi-Fi Alliance, anciennement WECA (*Wireless Ethernet Compatibility Alliance*), l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11.

Grâce au réseau WLAN, il est possible de créer des réseaux locaux sans fil pour peu que la station à connecter ne soit pas trop distante par rapport au point d'accès. Dans la pratique, le Wi-Fi permet de relier des ordinateurs portables, des machine de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit (11 Mbps ou plus) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert.

La norme 802.11 s'attache à définir les couches basses du modèle OSI pour une liaison sans fil utilisant des ondes électromagnétiques, c'est-à-dire :

- la **couche physique** (notée parfois *couche PHY*), proposant trois types de codage de l'information.
- la **couche liaison de données**, constitué de deux sous-couches : le contrôle de la liaison logique (**Logical Link Control**, ou **LLC**) et le contrôle d'accès au support (**Media Access Control**, ou **MAC**)

La couche physique définit la modulation des ondes radioélectriques 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. La norme 802.11 propose en réalité trois couches physiques, définissant des modes de transmission alternatifs :

Couche Liaison de données (MAC)	**802.2** 802.11
Couche Physique(PHY)	DSSS FHSS Infrarouges

Figure 1.1 – Couches basses du modèle OSI pour le sans fil

Il est possible d'utiliser n'importe quel protocole sur un réseau sans fil WiFi au même titre que sur un réseau Ethernet.

1.2. Les différentes normes WiFi :

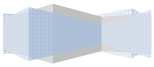
La norme *IEEE 802.11* est en réalité la norme initiale offrant des débits de 1 ou 2 Mbps. Des révisions ont été apportées à la norme originale afin d'optimiser le débit (c'est le cas des normes 802.11a, 802.11b et 802.11g, appelées normes 802.11 physiques) ou bien préciser des éléments afin d'assurer une meilleure sécurité ou une meilleure interopérabilité. Voici un tableau présentant les différentes révisions de la norme 802.11 et leur signification :

Nom de la norme	Nom	Description
802.11a	Wifi5	La norme 802.11a (baptisé *WiFi 5*) permet d'obtenir un haut débit (54 Mbps théoriques, 30 Mbps réels). La norme 802.11a spécifie 8 canaux radio dans la bande de fréquence des 5 GHz.
802.11b	Wifi	La norme 802.11b est la norme la plus répandue actuellement. Elle propose un débit

		théorique de 11 Mbps (6 Mbps réels) avec une portée pouvant aller jusqu'à 300 mètres dans un environnement dégagé. La plage de fréquence utilisée est la bande des 2.4 GHz, avec 3 canaux radio disponibles.
802.11c	Pontage 802.11 vers 802.1d	La norme 802.11c n'a pas d'intérêt pour le grand public. Il s'agit uniquement d'une modification de la norme 802.1d afin de pouvoir établir un pont avec les trames 802.11 (niveau *liaison de données*).
802.11d	Internationalisation	La norme 802.11d est un supplément à la norme 802.11 dont le but est de permettre une utilisation internationale des réseaux locaux 802.11. Elle consiste à permettre aux différents équipements d'échanger des informations sur les plages de fréquences et les puissances autorisées.
802.11e	Amélioration de la qualité de service	La norme 802.11e vise à donner des possibilités en matière de qualité de service au niveau de la couche *liaison de données*. Ainsi cette norme a pour but de définir les besoins des différents paquets en termes de bande passante et de retard de transmission de telle manière à permettre notamment une meilleure transmission de la voix et de la vidéo.
802.11f	Itinérance (roaming)	La norme 802.11f est une recommandation à l'intention des vendeurs de point d'accès pour une meilleure interopérabilité des produits. Elle propose le protocole *Inter-Access point roaming protocol* permettant à un utilisateur itinérant de changer de point d'accès de façon transparente lors d'un déplacement, quelles que soient les marques des points d'accès présentes

		dans l'infrastructure réseau. Cette possibilité est appelée *itinérance* (ou *roaming en anglais*)
802.11g		La norme 802.11g offre un haut débit (54 Mbps théoriques, 30 Mbps réels) sur la bande de fréquence des 2.4 GHz. La norme 802.11g a une compatibilité ascendante avec la norme 802.11b, ce qui signifie que des matériels conformes à la norme 802.11g peuvent fonctionner en 802.11b
802.11h		La norme *802.11h* vise à rapprocher la norme 802.11 du standard Européen (HiperLAN 2, d'où le *h* de 802.11h) et être en conformité avec la réglementation européenne en matière de fréquence et d'économie d'énergie.
802.11i		La norme *802.11i* a pour but d'améliorer la sécurité des transmissions (gestion et distribution des clés, chiffrement et authentification). Cette norme s'appuie sur l'*AES* (*Advanced Encryption Standard*) et propose un chiffrement des communications pour les transmissions utilisant les technologies 802.11a, 802.11b et 802.11g.
802.11Ir		La norme *802.11r* a été élaborée de telle manière à utiliser des signaux infrarouges. Cette norme est désormais dépassée techniquement.
802.11j		La norme *802.11j* est à la réglementation japonaise ce que le 802.11h est à la réglementation européenne.

Tableau 1.1 : différentes révisions de la norme 802.11 et leur signification



1.3. Modes de déploiement des réseaux sans fil WiFi :

Il existe différents types d'équipement pour la mise en place d'un réseau sans fil Wifi :

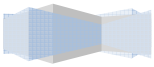
- Les **adaptateurs sans fil** ou **cartes d'accès** (en anglais *wireless adapters* ou *network interface controller*, noté *NIC*) : il s'agit d'une carte réseau à la norme 802.11 permettant à une machine de se connecter à un réseau sans fil. Les adaptateurs WiFi sont disponibles dans de nombreux formats (carte **PCI**, carte **PCMCIA**, adaptateur **USB**, carte **CompactFlash**, ...). On appelle **station** tout équipement possédant une telle carte.
- Les **points d'accès** (notés **AP** pour *Access point*, parfois appelés *bornes sans fil*) permettant de donner un accès au réseau filaire (auquel il est raccordé) aux différentes stations avoisinantes équipées de cartes wifi.

Le standard 802.11 définit deux modes opératoires :

- Le **mode infrastructure** dans lequel les clients sans fil 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 (notée **STA**) se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situées dans sa zone de couverture est appelé *ensemble de services de base* (en anglais *basic service set*, noté **BSS**) et constitue une cellule. Chaque *BSS* est identifié par un *BSSID*, un identifiant de 6 octets (48 bits). En mode *infrastructure*, le *BSSID* correspond à l'**adresse MAC** du point d'accès.



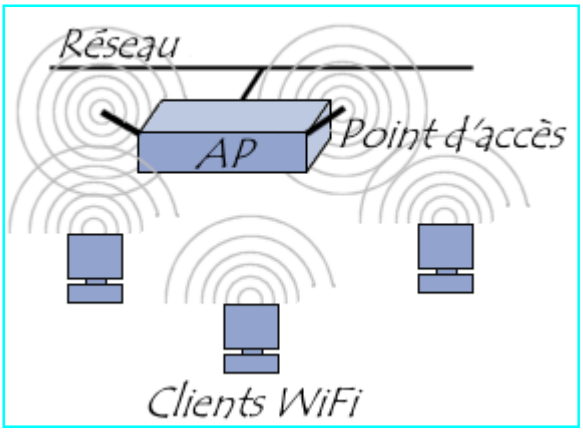


Figure 1.2 – mode infrastructure d'un réseau sans fil

Il est possible de relier plusieurs points d'accès entre eux (ou plus exactement plusieurs *BSS*) par une liaison appelée *système de distribution* (notée **DS** pour *Distribution System*) afin de constituer un *ensemble de services étendu* (*extended service set* ou *ESS*). Le système de distribution (*DS*) peut être aussi bien un réseau filaire, qu'un câble entre deux points d'accès ou bien même un réseau sans fil !

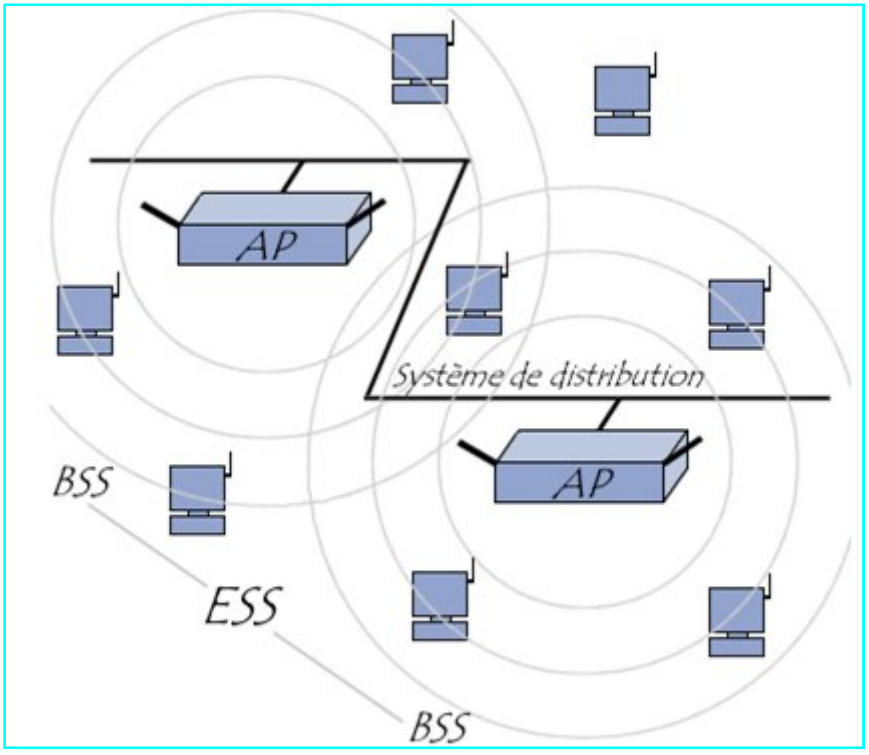


Figure 1.3 – Architecture d'un ESS

Un *ESS* est repéré par un **ESSID** (*Service Set Identifier*), c'est-à-dire un identifiant de 32 caractères de long (au format *ASCII*) servant de nom pour le réseau. L'*ESSID*, souvent abrégé

en **SSID**, représente le nom du réseau et représente en quelque sorte un premier niveau de sécurité dans la mesure où la connaissance du **SSID** est nécessaire pour qu'une station se connecte au réseau étendu.

Lorsqu'un utilisateur nomade passe d'un *BSS* à un autre lors de son déplacement au sein de l'*ESS*, l'adaptateur réseau sans fil de sa machine est capable de changer de point d'accès selon la qualité de réception des signaux provenant des différents points d'accès. Les points d'accès communiquent entre eux grâce au système de distribution afin d'échanger des informations sur les stations et permettre le cas échéant de transmettre les données des stations mobiles. Cette caractéristique permettant aux stations de "passer de façon transparente" d'un point d'accès à un autre est appelé *itinérance* (en anglais **roaming**).

Lors de l'entrée d'une station dans une cellule, celle-ci diffuse sur chaque canal une requête de sondage (*probe request*) contenant l'*ESSID* pour lequel elle est configurée ainsi que les débits que son adaptateur sans fil supporte. Si aucun *ESSID* n'est configuré, la station écoute le réseau à la recherche d'un *SSID*.

En effet, chaque point d'accès diffuse régulièrement (à raison d'un envoi toutes les 0.1 secondes environ) une **trame balise** (nommée **beacon** en anglais) donnant des informations sur son *BSSID*, ses caractéristiques et éventuellement son *ESSID*. L'*ESSID* est automatiquement diffusé par défaut, mais il est possible de désactiver cette option.

A chaque requête de sondage reçue, le point d'accès vérifie l'*ESSID* et la demande de débit présent dans la *trame balise*. Si l'*ESSID* correspond à celui du point d'accès, ce dernier envoie une réponse contenant des informations sur sa charge et des données de synchronisation. La station recevant la réponse peut ainsi constater la qualité du signal émis par le point d'accès afin de juger de la distance à laquelle il se situe. En effet, d'une manière générale, plus un point d'accès est proche, meilleur sera le débit.

Une station se trouvant à la portée de plusieurs points d'accès (possédant bien évidemment le même *SSID*) pourra ainsi **choisir** le point d'accès offrant le meilleur compromis de débit et de charge.

Lorsqu'une station se trouve dans le rayon d'action de plusieurs points d'accès, c'est elle qui choisit auquel se connecter !

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 (*peer to peer* en anglais), c'est-à-dire un réseau dans lequel chaque machine joue en même temps le rôle de client et le rôle de point d'accès.

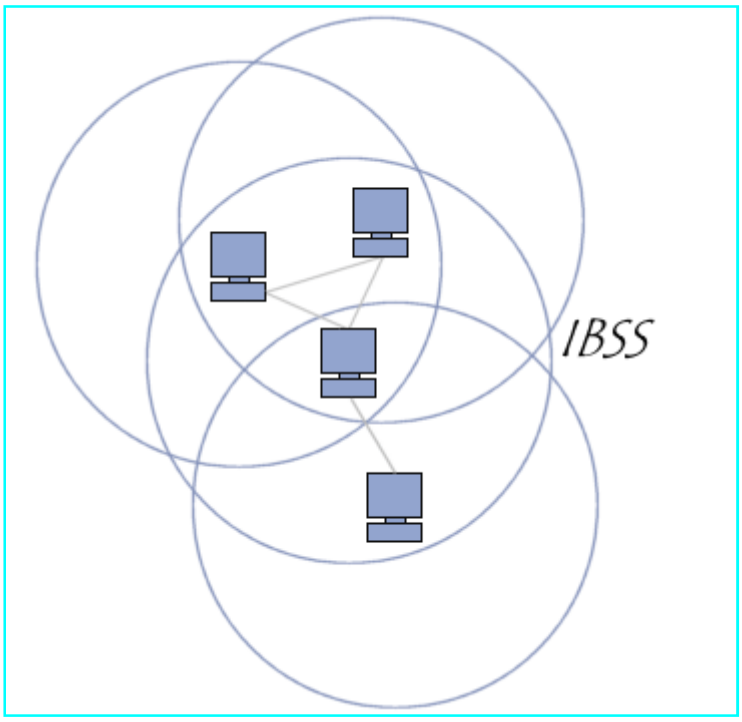


Figure 1.4 – Mode Ad Hoc d'un réseau sans fil

L'ensemble formé par les différentes stations est appelé *ensemble de services de base indépendants* (en anglais **independant basic service set**, abrégé en *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. Il est identifié par un *SSID*, comme l'est un *ESS* en mode infrastructure.

Dans un réseau ad hoc, la portée du *BSS indépendant* est déterminée par la portée de chaque station. Cela signifie que si deux des stations du réseau sont hors de portée l'une de l'autre, elles ne pourront pas communiquer, même si elles "voient" d'autres stations. En effet, contrairement au mode infrastructure, le mode *ad hoc* ne propose pas de *système de*

distribution capable de transmettre les trames d'une station à une autre. Ainsi un *IBSS* est par définition un réseau sans fil restreint.

Concernent notre cas au sein de l'Ecole Nationale Polytechnique, le mode infrastructure est le mode idéal permettant aux différents utilisateurs d'accéder via le sans fil aussi bien aux ressources de l'Ecole qu'à internet.

Ce mode a été choisis pour pouvoir étendre le réseau déjà existant**,** partager la connexion Internet par le sans fil, et éventuellement avoir une zone de couverture fixe qui ne dépende pas des stations présentes.

1.4. Sécurité dans les réseaux sans fil (WiFi) :

La sécurité d'un réseau est un niveau de garantie que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs des dites machines possèdent uniquement les droits qui leur ont été octroyés.

Il peut s'agir :

— d'empêcher des personnes non autorisées d'agir sur le système de façon malveillante ;

— d'empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au système.

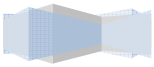
Pour rendre sûr un réseau sans fil, on distingue généralement les trois approches suivantes :

— intervention au niveau des protocoles dans les nœuds sans fil eux-mêmes. C'est l'approche sécurité au niveau 2 : niveau MAC (Medium Access Control) ;

— adaptation pour les réseaux sans fil de l'approche maintenant classique du réseau privé virtuel. Cette approche est la plus souvent implémentée au niveau réseau ;

— sécurisation des applications utilisées au niveau transport ou au niveau application.

La sécurité est un vaste domaine et l'arrivée des réseaux sans fil ne fait que compliquer les problèmes. Il ne nous est pas possible de présenter une analyse exhaustive du sujet dans le cadre de notre projet de fin d'études. Nous insisterons donc sur les principales solutions qui ont été développées, notamment au niveau de la couche application du modèle OSI.



1.5. Comment rendre sûr un réseau sans fil au niveau applicatif:

La sécurité d'un réseau sans fil peut être obtenue au-dessus du niveau réseau en sécurisant les applications. Les solutions possibles à ce niveau sont :

1.5.1. Protocoles :

Les protocoles qui permettent de sécuriser les applications sont les suivants :

— SSL (Secure Sockets Layer), développé par Netscape pour fournir de la sécurité aux applications de navigation sur le Web et qui offre des services d'authentification, de chiffrement et d'intégrité ;

— TSL (Transport Security Layer) qui est très proche de SSL et offre une bonne gestion de la clé, laquelle est régénérée à chaque nouvelle association entre la station et le point d'accès ;

— SSH (Secure Shell) qui permet d'ouvrir une connexion sécurisée entre deux machines distantes et est particulièrement adapté pour ouvrir la connexion vers des machines de type Unix.

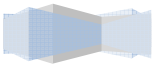
— PGP (Pretty Good Privacy), un système de sécurité plus particulièrement destiné au courrier électronique.

Ces solutions de sécurité ne vont pas permettre de lutter contre les intrusions dans le réseau sans fil. Pour cela, on pourra envisager l'utilisation de pare-feu. C'est ce que nous allons décrire maintenant.

1.5.2. Pare-feu :

Les pare-feu, ou firewalls, sont des outils classiques de sécurité, qui permettent de protéger un équipement ou un réseau des attaques extérieures. Le mode de fonctionnement d'un pare-feu est le filtrage des paquets par adresse et/ou par port. Cela permet de contrôler à l'entrée d'un système les sources et les applications portées par les paquets provenant de l'extérieur.

Le pare-feu se place généralement au point de contact entre l'équipement et le réseau que l'on souhaite protéger. Dans le cas d'un réseau sans fil, il faudrait mettre un pare-feu sur chaque point d'accès ou adopter l'architecture illustrée par la figure **1.5**.



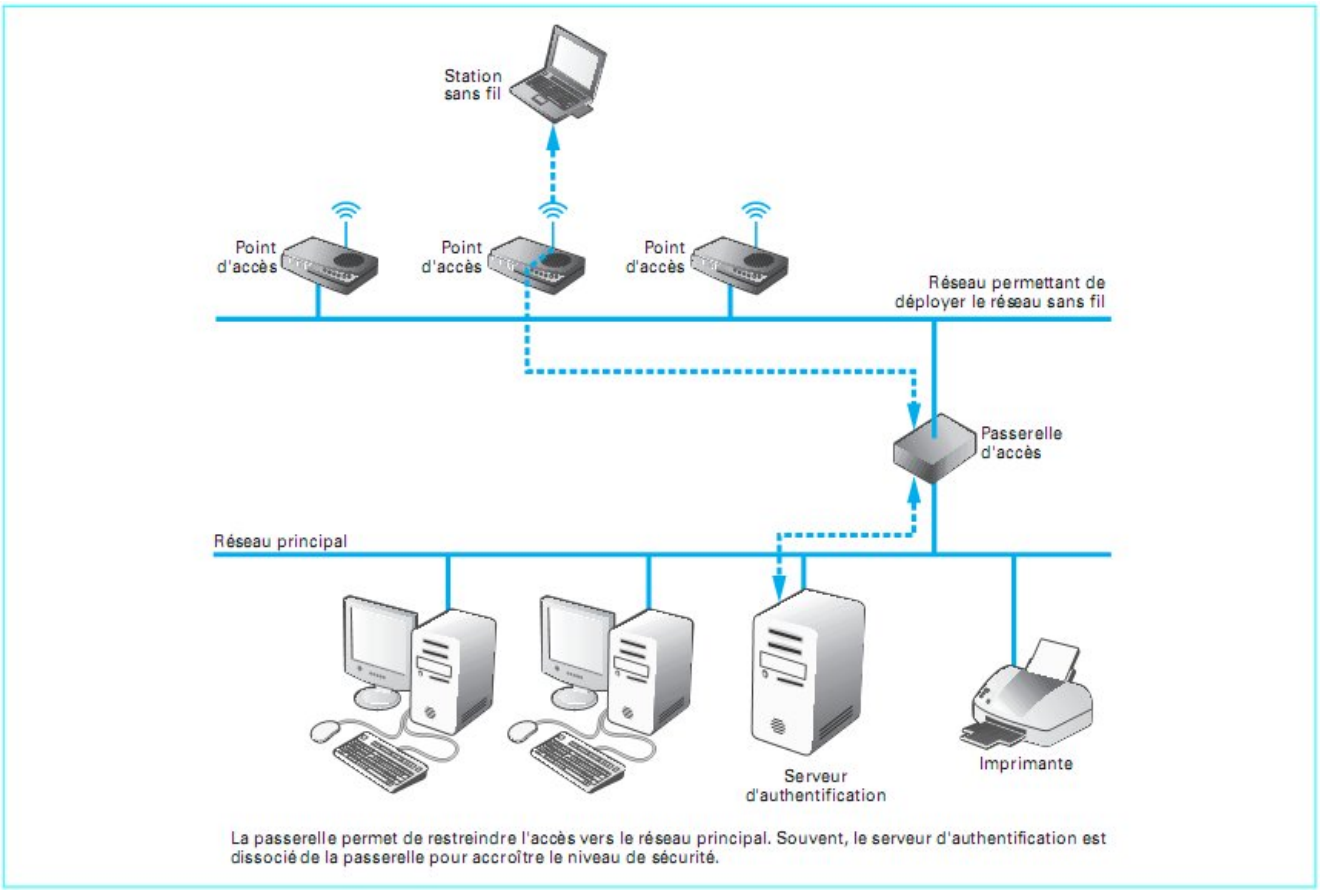


Figure 1.5 – Passerelle permettant de sécuriser l'accès au réseau principal

1.5.3. Identification :

Afin d'initier à la notion d'identité, nous allons présenter deux solutions possibles à réaliser, à savoir le Single Signe On (SSO) et le portail captif.

1.5.3.1. SSO : [2]

L'**authentification unique** (ou identification unique ; en anglais **Single Sign-On** ou **SSO**) est une méthode permettant à un utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques (ou sites internet sécurisés).

a) Objectifs :

Les objectifs sont multiples :

— Simplifier pour l'utilisateur la gestion de ses mots de passe : plus l'utilisateur doit gérer de mots de passe, plus il aura tendance à utiliser des mots de passe similaires ou simples à mémoriser, abaissant par la même occasion le niveau de sécurité que ces mots de passe offrent ;

— Simplifier la gestion des données personnelles détenues par les différents services en ligne, en les coordonnant par des mécanismes de type méta-annuaire ;

— Simplifier la définition et la mise en œuvre de politiques de sécurité. Il existe trois grandes classes d'approches pour la mise en œuvre de systèmes d'authentification unique : les approches centralisées, les approches fédératives et les approches coopératives.

b) Avantages :

Les avantages de l'authentification unique incluent :

— La réduction de la fatigue de mot de passe : manque de souplesse liée à l'utilisation de différentes combinaisons de nom d'utilisateur et de mot de passe;

— La réduction du temps passé à saisir le même mot de passe pour le même compte ;

— La réduction du temps passé en support informatique pour des oublis de mots de passe;

— La centralisation des systèmes d'authentification ;

— La sécurisation à tous les niveaux d'entrée/de sortie/d'accès aux systèmes sans sollicitation multiple des utilisateurs ;

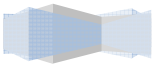
— La centralisation des informations de contrôles d'accès pour les tests de conformités aux différentes normes.

Les technologies fournissant SSO utilisent des serveurs centralisés d'authentification que toutes les autres applications et systèmes utilisent pour l'authentification, combinant ceci avec des techniques logicielles pour s'assurer que les utilisateurs n'aient pas à entrer leurs identifiants plus d'une fois.

c) Architecture :

Approche centralisée :

Le principe de base est ici de disposer d'une base de données globale et centralisée de tous les utilisateurs ou d'un annuaire. Cela permet également de centraliser la gestion de la politique de sécurité.



Approche fédérative :

Dans cette approche, chaque service gère une partie des données d'un utilisateur (l'utilisateur peut donc disposer de plusieurs comptes), mais partage les informations dont il dispose sur l'utilisateur avec les services partenaires.

Cette approche a été développée pour répondre à un besoin de gestion décentralisée des utilisateurs, où chaque service partenaire désire conserver la maîtrise de sa propre politique de sécurité, comme par exemple un ensemble d'universités indépendantes d'un point de vue organisationnel.

Approche coopérative :

L'approche coopérative, dont les systèmes Shibboleth est le principal représentant, part du principe que chaque utilisateur dépend d'une des entités partenaires. Ainsi, lorsqu'il cherche à accéder à un service du réseau, l'utilisateur est authentifié par le partenaire dont il dépend. Comme dans l'approche fédérative, cependant, chaque service du réseau gère indépendamment sa propre politique de sécurité.

1.5.3.2. Portail Captif : [3]

Le portail captif est une solution d'authentification qui ne nécessite pas d'installation particulière sur le client. L'authentification est basée sur une page web sécurisée dans laquelle l'utilisateur donne son identifiant (login) et son mot de passe. Il est alors authentifié sur le réseau.

La figure **1.6** donne le schéma de ce mécanisme. Le client Wi-Fi est associé avec le point d'accès et récupère une adresse IP par le serveur DHCP. Lors de la première demande de connexion http vers l'extérieur, l'utilisateur est renvoyé automatiquement vers la page d'authentification du portail captif (en https). Il saisit ensuite son identifiant et son mot de passe. Si le firewall valide cette authentification, l'utilisateur est redirigé vers la page qu'il avait choisi au départ (en http).

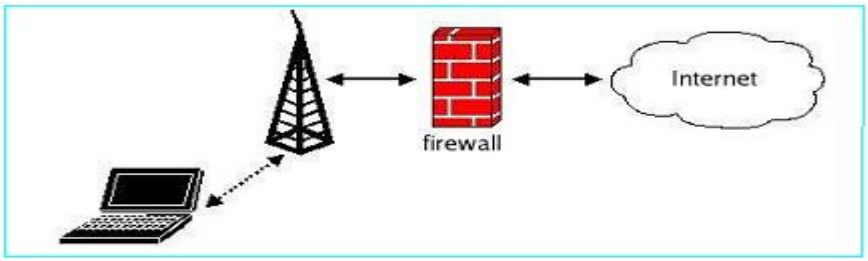


Figure 1.6 – Le portail captif hébergé par le firewall permet (après authentification) de sortir sur Internet

Un ESSID unique est utilisé par toutes les communautés d'utilisateurs afin de se connecter au Wi-Fi. La base des utilisateurs est stockée dans un annuaire LDAP.

Sur les ordinateurs clients, il suffit de définir les paramètres IP en DHCP et d'entrer l'ESSID voulu. L'ordinateur est alors prêt à se connecter. Il faut ensuite entrer l'utilisateur et son mot de passe au niveau du firewall. Et c'est fini, l'utilisateur peut immédiatement se connecter.

Dès qu'il active la carte Wi-Fi, l'utilisateur reçoit ses paramètres IP du serveur DHCP du firewall. Dès qu'il lance son navigateur sur une page (enp.edu.dz par exemple), il est automatiquement redirigé vers la page d'accueil de l'authentification. Il entre alors son login (sécurisé par https). Puis il saisit son mot de passe (toujours sécurisé par https).

Dès que l'authentification est réussie, l'utilisateur est dirigé automatiquement vers la page qu'il avait demandée au départ.

1.6. Réseau wifi dans un campus universitaire :

1.6.1. Objectifs :

La nécessité d'étendre le réseau filaire au niveau du campus universitaire et donner une liberté supplémentaire aux étudiants et aux personnels de l'institution, amène à entreprendre le déploiement d'un réseau sans fil wifi.

La mise en place d'un tel réseau passe par plusieurs étapes, la première se fera à travers l'étude de l'architecture de l'Ecole afin de définir les régions où l'activité est susceptible d'être élevée, cela permet de positionner convenablement les antennes, par la suite, il faut définir comment doit être relié le réseau sans fil avec le réseau filaire.

En fin, il est nécessaire de fournir aux utilisateurs du sans-fil une sécurité d'accès et de transmission.

1.6.2. Déploiement des Antennes :

L'installation des antennes wifi n'est pas toujours évidente, surtout dans un établissement où la surface à couvrir peut être très grande, de plus les conditions environnementales susceptibles de perturber les transmissions sans fil doivent être prises en considération.

1.6.3. Interconnexion entre le réseau sans fil et le réseau filaire :

Un réseau sans fil est généralement relié à un réseau filaire, cela donne lieu à plusieurs types d'interconnexions.

La première qu'on peut citer est celle où on utilise une liaison directe par simple câble Ethernet.

Une deuxième approche, consiste à utiliser une passerelle pour pouvoir relier les deux réseaux et éventuellement faire du filtrage multi niveaux.

Enfin, cette liaison peut faire appel à des VLAN (Virtuel Local Area Network) pour être établi.

1.6.4. Sécurité dans le réseau sans fil :

Le problème de la sécurité dans les réseaux sans fil au sein d'un campus universitaire est un enjeu majeur, car il est à noter qu'un réseau non sécurisé, ouvre des possibilités d'intrusions à des utilisateurs non admis au sein de l'établissement, de plus il représente une opportunité pour des pirates de détourner des informations qui peuvent être d'une importance capitale dans certains cas.

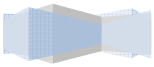
L'intégration de la sécurité peut être envisagée sous deux aspects :

Le premier étant de mettre en place une politique de sécurité au niveau de la couche physique édictée par la normalisation.

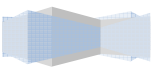
Le deuxième aspect est situé au niveau de la couche application. Dans cette dernière, plusieurs solutions peuvent être mises en place, dont celle qui se base sur la « gestion d'identité » où chaque utilisateur doit s'identifier avec des clefs appropriées.

1.7. Conclusion :

La démocratisation des réseaux Wi-Fi a grandement simplifié le déploiement des infrastructures universitaires. Même si cette évolution est extrêmement positive, gardons-nous de penser que cette technologie est exempte de défauts. Contrairement aux réseaux filaires qui ne peuvent être attaqués que de façon distante, depuis Internet, un réseau sans fil pourra toujours être pénétré localement. Aussi, pour ne conserver que les bénéfices du Wi-Fi, il est fortement recommandé de mettre en place une stratégie de sécurisation.



À l'issue de ce chapitre, l'intégration de la sécurité au niveau applicatif du modèle OSI propose diverses solutions dont la gestion des identités. Cette gestion d'identité peut être répartie. Couplée à la mobilité du personnel universitaire aussi bien étudiant qu'enseignant, la fédération d'identité se présente comme une solution plus qu'adaptée au déploiement d'un service d'accès au réseau sans fil permettant ainsi de faciliter la gestion répartie de l'identité.

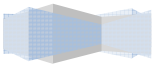


Chapitre 2 :

Fédération d'identité

2.1. Introduction :

Ce chapitre va présenter la fédération d'identité, ainsi nous allons introduire Shibboleth qui va être la brique sur laquelle cette fédération repose.



2.2. Fédération d'identité :

L'objectif de la fédération d'identités est de faciliter le partage de ressources numériques en ligne entre établissements d'enseignement supérieur en interconnectant leurs services d'authentification. Il devient possible d'ouvrir l'accès à une ressource numérique (pédagogique, scientifique, etc.) à une population identifiée, sans devoir gérer localement l'enregistrement des utilisateurs. Exemple d'usage : l'Ecole Nationale Polytechnique ouvre l'accès à un cours en ligne d'électronique, mais uniquement aux étudiants de cette discipline appartenant à l'Ecole. Un étudiant en électronique d'une autre université pourra ainsi accéder au cours en ligne en s'authentifiant sur le site de son université et sans que l'Ecole ne doit l'enregistrer en tant qu'utilisateur. Une coopération entre institutions est alors impérative.

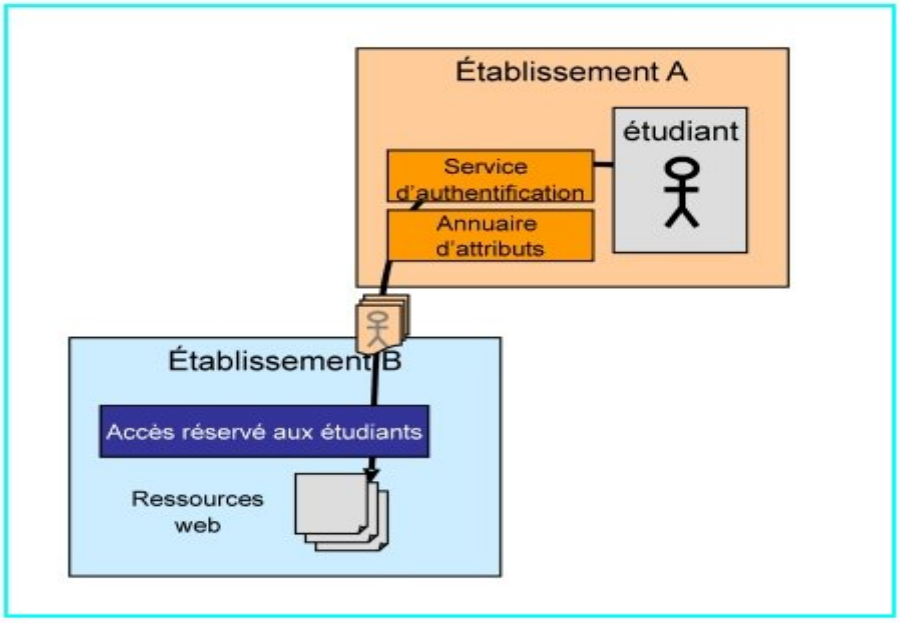


Figure 2.1 – Partage des ressources en ligne à travers la fédération d'identité

2.2.1. La fédération, un cercle de confiance :

La fédération concrétise, pour un groupement d'établissements d'enseignement supérieur, l'interconnexion de leurs services d'authentification et l'utilisation d'un ensemble commun d'attributs utilisateurs. Un établissement qui gère un ensemble d'utilisateurs est appelé fournisseur d'identités. Un fournisseur de services est une entité - établissement, administration, société privée - qui propose une ressource numérique en ligne au sein de la fédération. Techniquement, les relations de confiance entre les membres d'une fédération reposent sur des certificats électroniques et des métadonnées partagées. En outre la confiance s'établit administrativement entre les participants de la fédération au travers d'une convention.

Un même établissement peut participer à plusieurs fédérations, il peut également jouer à la fois le rôle de fournisseur d'identités et de fournisseur de services.

2.2.2. Principes de la propagation d'identités et d'attributs :

L'objectif de la propagation d'identités est double : déléguer l'authentification à l'établissement d'origine de l'utilisateur et obtenir certains attributs de l'utilisateur (pour gérer le contrôle d'accès ou personnaliser les contenus).

La délégation de l'authentification réutilise les techniques de Single Sign-On (redirection, cookies...). Lors de l'accès initial à une ressource numérique, l'utilisateur est redirigé vers le service de découverte de la fédération, d'où il sélectionne son établissement d'origine ; il est ensuite renvoyé vers son fournisseur d'identités. Le pré requis pour le fournisseur d'identités est de disposer d'un service d'authentification globale.

A l'issue de la phase d'authentification, le fournisseur de services prend connaissance de l'identifiant de l'utilisateur qui lui permettra, lors d'une deuxième phase, d'obtenir les attributs de l'utilisateur. Le fournisseur d'identité a la possibilité de définir, de façon différenciée pour chaque interlocuteur, quels attributs utilisateur pourront être dévoilés.

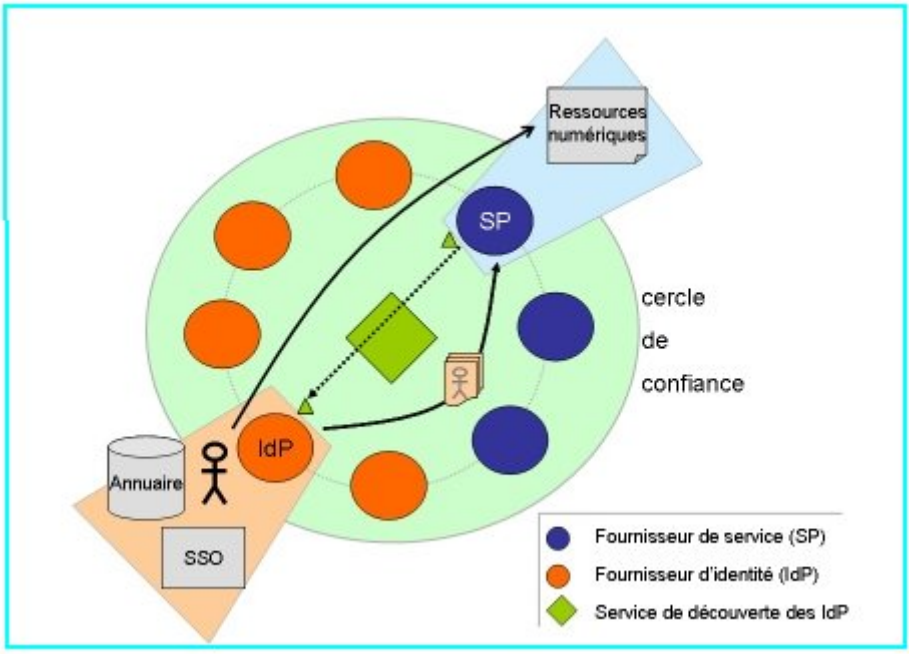


Figure 2.2 – Principes de la propagation d'identités et d'attributs

2.2.3. Shibboleth, le produit utilisé pour gérer la fédération

Shibboleth est une application bien adaptée au contexte universitaire. Cette application est la pierre angulaire de nombreuses fédérations académiques. Shibboleth est une application Open Source, développée en Java. Elle améliore la sécurité de l'accès aux applications extérieures. De nombreuses applications supportent nativement Shibboleth.

2.3. SAML :

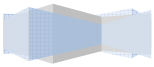
SAML (*Security Assertion Markup Language* [4]) est un ensemble de spécifications qui définissent comment des services peuvent s'échanger des assertions de sécurité (authentification, autorisation, attributs), indépendamment des technologies utilisées par chacun de ces services (PKI [5], SSO, LDAP, etc.). SAML ne couvre donc pas tout le spectre de la gestion des identités, par exemple ne définit pas de protocole de SSO ou une sémantique d'attributs standard. Il s'appuie sur des standards préexistants (XML, SSL, etc.) et a été conçu avec suffisamment d'abstraction pour rendre inter opérable des systèmes hétérogènes, et s'articuler au mieux avec d'autres mécanismes de gestion d'identités.

SAML est constitué de différents protocoles, qui correspondent aux différents cas d'usage adressés par ce standard. Un protocole SAML décrit de façon abstraite comment une entité interagit avec un système SAML, généralement sous la forme d'une séquence de requêtes et de réponses. Un *protocol binding* est la traduction d'un tel protocole abstrait en un protocole de communication implémentable informatiquement, par exemple sous la forme de *Web Services* SOAP [6]. De plus, SAML étant très abstrait pour assurer l'interopérabilité des systèmes (notamment sur la composition des messages), il existe des « profils SAML » qui restreignent (ou étendent) la variabilité d'un protocole de base pour des usages particuliers. En s'accordant sur l'utilisation d'un certain profil, deux entités voulant communiquer en SAML se simplifient l'interopérabilité.

Voici un exemple d'assertion (d'authentification SAML) :

```
<saml:Assertion
   MajorVersion="1"
   MinorVersion="0"
   AssertionID="128.9.167.32.12345678"
   Issuer="Comite Reseau des Universites"
   IssueInstant="2009-06-21T10:02:00Z">

   <saml:Conditions
```

```
    NotBefore="2009-06-21T10:02:00Z"
    NotAfter="2009-06-21T10:07:00Z" />

  <saml:AuthenticationStatement
    AuthenticationMethod="password"
    AuthenticationInstant="2009-06-21T10:02:00Z">
    <saml:Subject>

      <saml:NameIdentifier
        SecurityDomain="www.enp.edu.dz"
        Name="GACEM" />
    </saml:Subject>

  </saml:AuthenticationStatement>

</saml:Assertion>
```

Il s'agit d'un document XML signé et envoyé par un service A à un autre service, et qui dit en substance « cet utilisateur a été correctement authentifié par le service A ». SAML ne sert pas à authentifier l'utilisateur (n'importe quel système d'authentification peut être utilisé), mais à communiquer le fait qu'il a été correctement authentifié. L'assertion contient ici des informations sur la date et le mode d'authentification, et possède une durée de validité limitée.

SAML est déjà implémenté dans beaucoup de produits, et sert de fondation à d'autres normes, tel Shibboleth [7].

2.4. Shibboleth :

Shibboleth [8] désigne à la fois une norme et un produit (open source). C'est une mise en œuvre de SAML qui enrichit ses fonctionnalités de fédération d'identités en facilitant pour un ensemble de partenaires la mise en place de deux fonctionnalités importantes, la délégation d'authentification et la propagation d'attributs. Shibboleth a été conçu pour répondre aux besoins des communautés de l'enseignement supérieur et est déjà utilisé dans plusieurs pays : Etats-Unis, Angleterre, Suisse, Finlande, etc.

2.4.1. Le système Shibboleth :

L'objectif de cette partie est de montrer les interactions entre les acteurs du système qui permettent la délégation de l'authentification et la propagation des attributs utilisateurs.

2.4.1.1. Les acteurs du système :

L'architecture générale de Shibboleth peut être décrite comme suit :

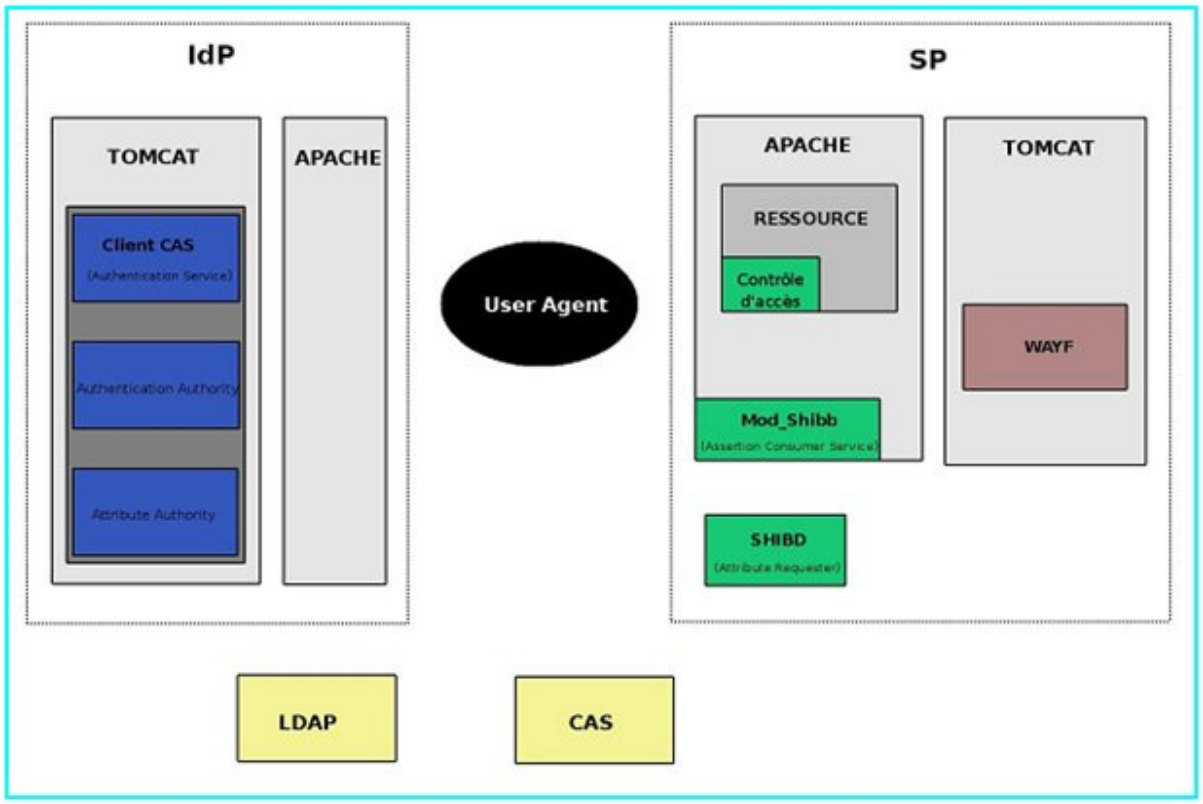


Figure 2.3 – Constituants de la solution Shibboleth

On distingue ainsi 4 acteurs : le navigateur (User Agent), le fournisseur de service (SP ou Service Provider), le fournisseur d'identité (IdP ou Identity Provider) et le service de découverte (WAYF ou Where Are You From).

a) Le navigateur (User Agent) :

Shibboleth répond à la problématique des applications web. Le premier acteur de l'architecture Shibboleth est donc logiquement le navigateur de l'utilisateur. Le navigateur doit répondre aux exigences habituelles en matière de navigation web, notamment l'interprétation des codes de retour HTTP (redirections [8]), ainsi que l'acceptation et la transmission des *cookies* selon les normes en vigueur [9] (cela vaut pour la très grande majorité des navigateurs web du marché).

b) Le fournisseur de services (Service Provider ou SP) :

Une entité proposant des ressources web sur la base d'un contexte de sécurité SAML est appelée « fournisseur de services » (ou *Service Provider)*, et sera par la suite nommée SP.

Le fournisseur de ressource a en particulier la charge de donner ou non l'accès aux ressources, en fonction des attributs utilisateur.

Notons dès à présent que les ressources web sont à *priori* quelconques ; on trouvera ainsi :

– Des **applicatifs web**, dont le contrôle d'accès peut être effectué indifféremment par un pré-filtre (tel un module Apache) ou au sein même de la logique applicative. Ces applicatifs pourront également utiliser les attributs utilisateur à d'autres fins que le contrôle d'accès (la personnalisation de l'interface ou la définition des rôles des utilisateurs par exemple) ;
– Des **pages statiques**, dont le contrôle d'accès doit être effectué par un pré-filtre.

c) Le fournisseur d'identités (Identity Provider ou IdP) :

Une entité authentifiant les utilisateurs et fournissant leurs attributs est appelée « fournisseur d'identités » (ou *Identity Provider*) et sera par la suite notée IdP.
Le fournisseur d'identités s'appuie sur le SI (Système d'Information) de l'établissement, tant pour l'authentification que pour la récupération des attributs utilisateur à propager. De ce fait, il est en général situé au plus proche du SI.

d) Le WAYF :

Le WAYF (pour *Where Are You From?*, « d'où êtes-vous ? ») est un service dont le but est d'orienter l'utilisateur vers son IdP.

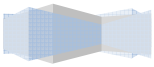
2.4.2. Déploiement de Shibboleth :

Déployer Shibboleth au sein de notre établissement nécessite de choisir une des architectures possibles, à savoir :
– Architecture Shibboleth sans SSO dans une mono fédération ;
– Architecture Shibboleth avec SSO dans une mono fédération ;
– Architecture Shibboleth avec SSO et WAYF (plusieurs établissements).
Comme aborder dans le chapitre 2, la technique SSO est parmi les méthodes proposées au niveau applicatif du modèle OSI pour sécuriser un réseau sans fil, dans notre cas (un seul établissement), la meilleure solution est de déployer Shibboleth avec SSO.

2.4.3. Le fonctionnement de Shibboleth avec SSO :

2.4.3.1. Première requête vers un SP :

Dans le modèle de CAS (*Central Authentication System)*, l'authentification n'est pas directement prise en charge par le service d'authentification de l'IdP ; celui-ci ne fait que rediriger le navigateur vers le serveur de SSO, qui renvoie alors à l'utilisateur un formulaire d'authentification (cf Figure **2.4**).



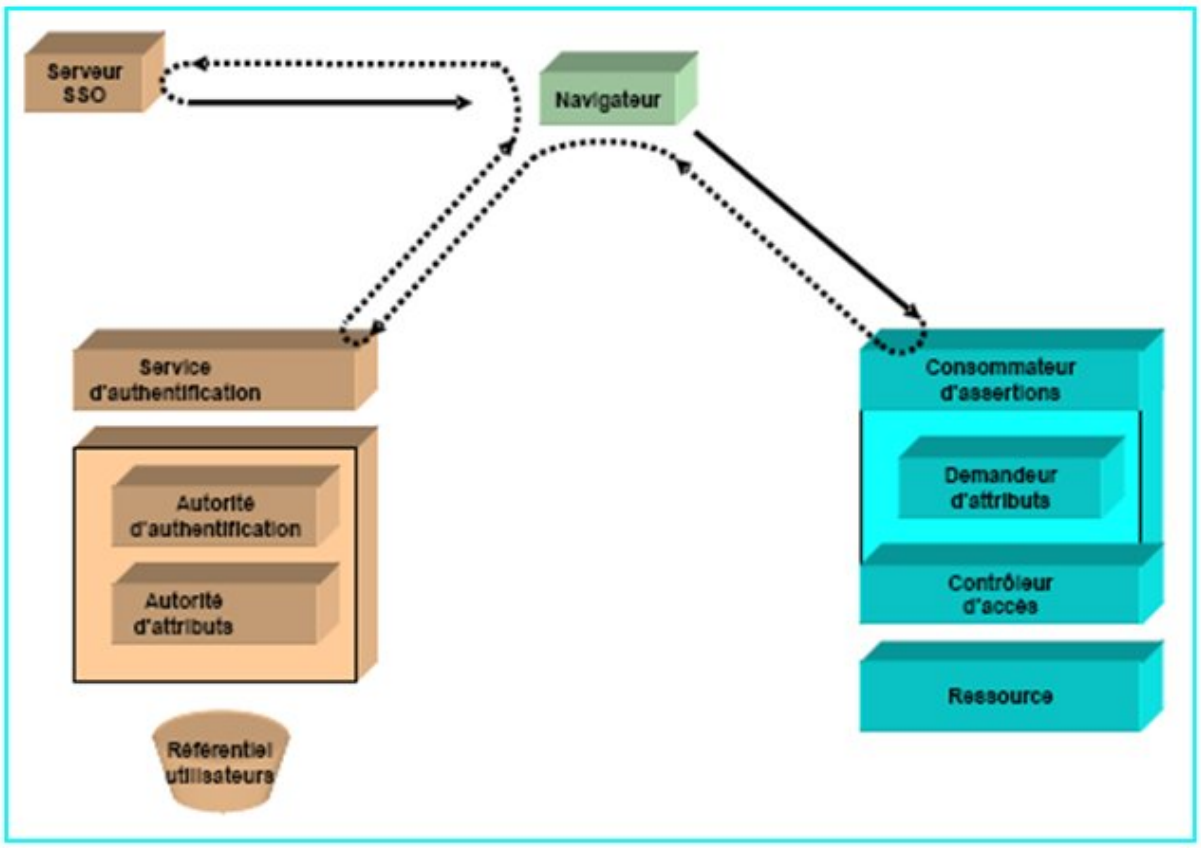


Figure 2.4 – Première requête à un SP dans un contexte SSO

Le navigateur remplit alors le formulaire et effectue une nouvelle requête vers le serveur SSO, qui le redirige vers l'IdP. Le service d'authentification de l'IdP, client SSO, effectue alors une nouvelle redirection du navigateur vers le SP, accompagné d'une assertion SAML. Cette assertion est signée par l'IdP, le SP pourra donc faire confiance à l'assertion. Elle contient un identifiant appelé *nameIdentifier*.

Cet identifiant est opaque, c'est-à-dire qu'il ne contient pas d'information personnelle concernant l'utilisateur. Il n'est utilisé que dans le cadre des échanges entre les différentes briques de Shibboleth, et n'est connu ni de la ressource accédée ni du SI (Système d'Information) de l'établissement. Un exemple de *nameIdentifier* est montré ci-dessous.

```
<saml:NameIdentifier
Format="urn:mace:shibboleth:1.0:nameIdentifier"
   NameQualifier="https://idp.example.org/shibboleth">
3f7b3dcf-1674-4ecd-92c8-1544f346baf8
</saml:NameIdentifier>
```

C'est cet identifiant opaque qui va permettre au SP de récupérer les attributs de l'utilisateur auprès de l'IdP. Les attributs de l'utilisateur sont transmis au SP par l'IdP, via l'appel d'un *Web Service*, et en échange du *nameIdentifier*.

Le SP peut alors effectuer le contrôle d'accès, éventuellement utiliser les attributs de l'utilisateur dans la logique applicative, puis retourner une réponse au navigateur.

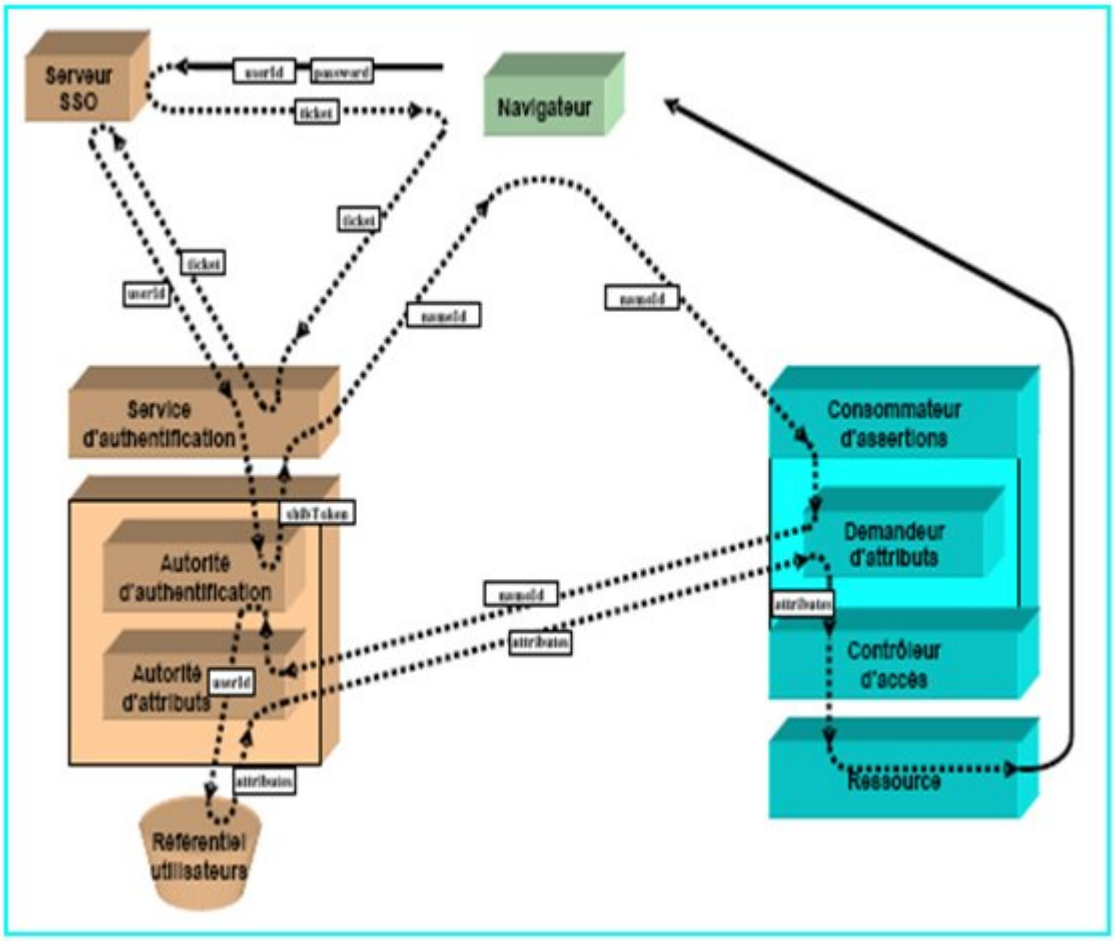


Figure 2.5 – Redirection vers un SP par le serveur SSO

Notons que, dans ce cas, l'identifiant de l'utilisateur n'est pas fourni par le navigateur, mais récupéré auprès du serveur CAS par l'IdP.

Pour résumer, du point de vue de l'utilisateur (cf Figure **2.6**), celui-ci :

– Effectue une requête auprès du SP (1) et reçoit une demande d'authentification du serveur SSO(2) ;

– S'authentifie auprès du serveur SSO (3) et reçoit une réponse du SP (4), qui donne accès ou non à la ressource demandée.

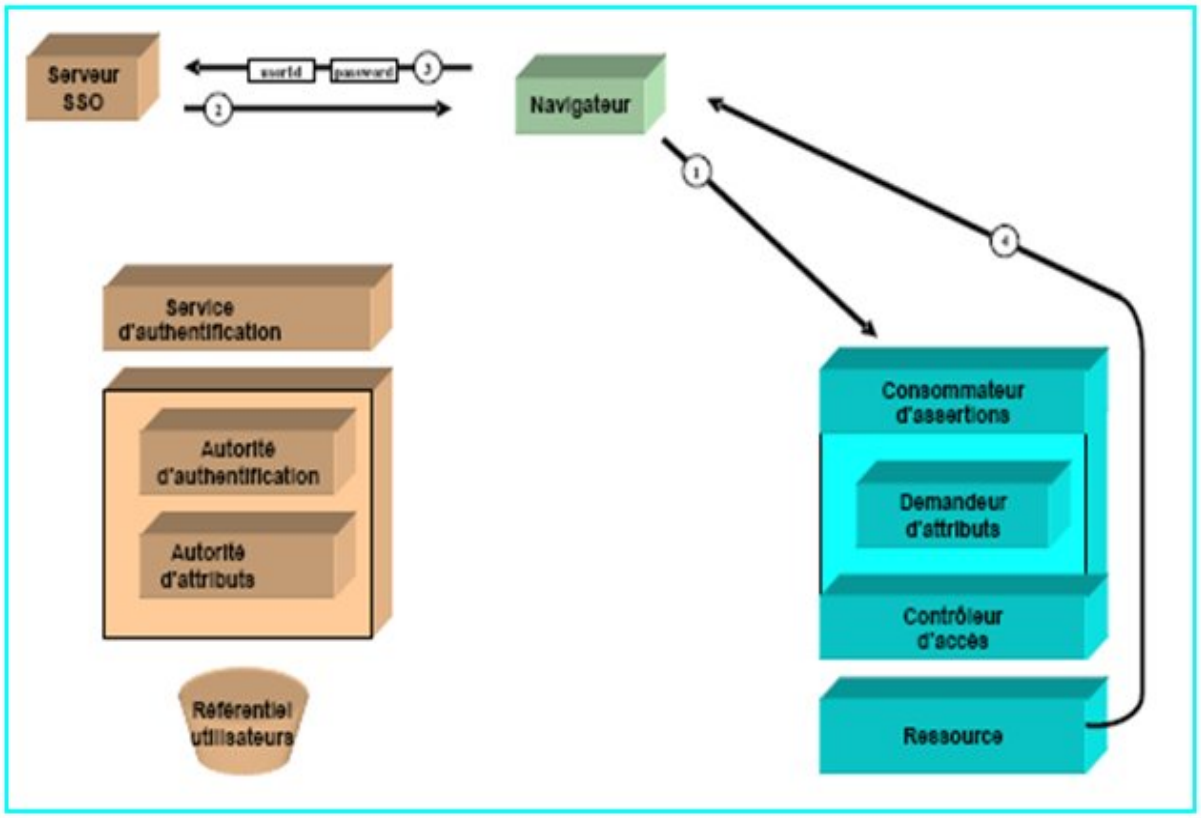


Figure 2.6 – Point de vue de l'utilisateur dans un contexte SSO

2.4.3.2. Requêtes suivantes au même SP :

Comme vu précédemment, une session étant mise en place entre le navigateur et le SP (en fait, le consommateur d'assertions du SP), ni l'IdP ni le serveur SSO n'interviennent plus par la suite pour l'accès au même SP (cf Figure **2.7**).

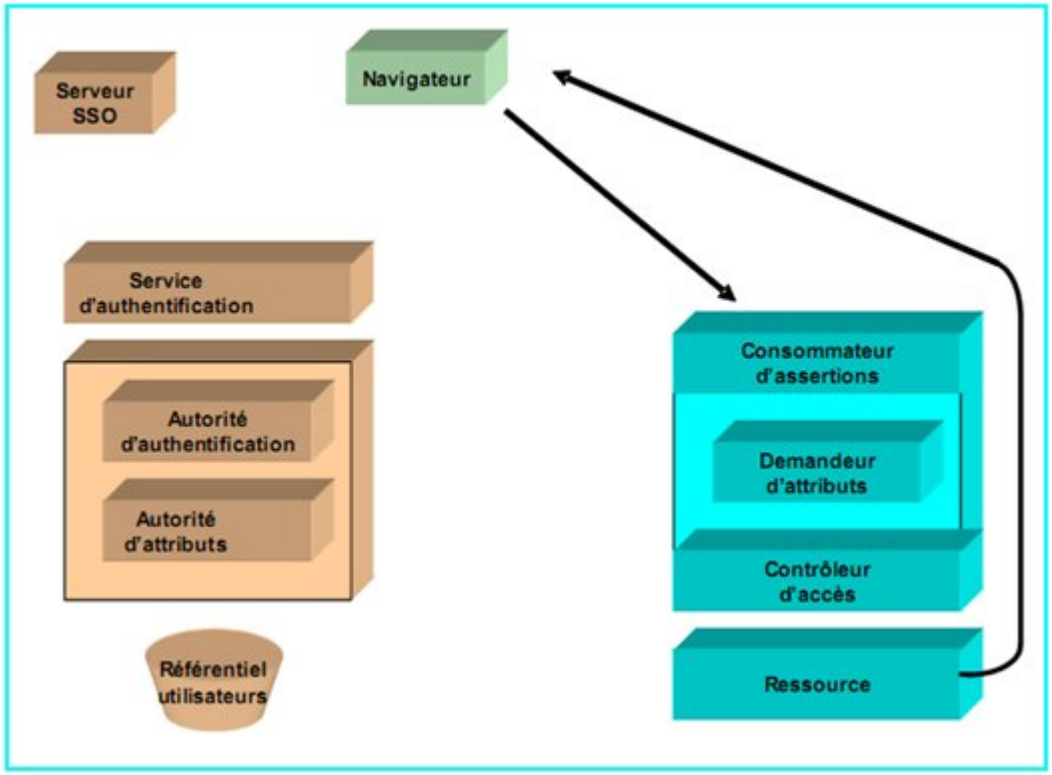


Figure 2.7 – Requêtes suivantes vers le même SP dans un contexte SSO

2.4.3.3. Requêtes suivantes vers un autre SP

Lorsque l'utilisateur est déjà authentifié auprès du serveur SSO, le navigateur dispose d'un identificateur de session qui lui permet de ne pas avoir à s'authentifier à nouveau (cf Figure **2.8**).

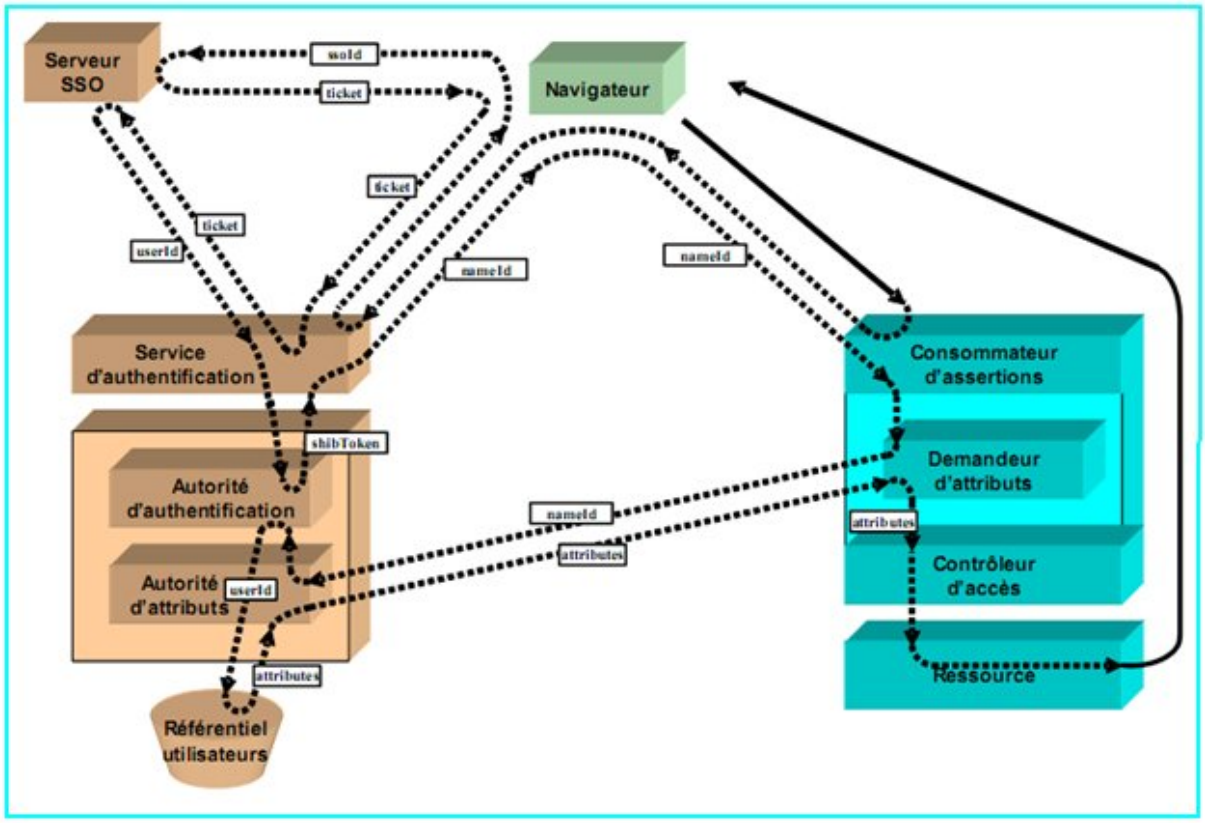


Figure 2.8 – Requêtes suivantes vers un autre SP dans un contexte SSO

L'authentification de l'utilisateur est alors complètement transparente (cf Figure **2.9**).

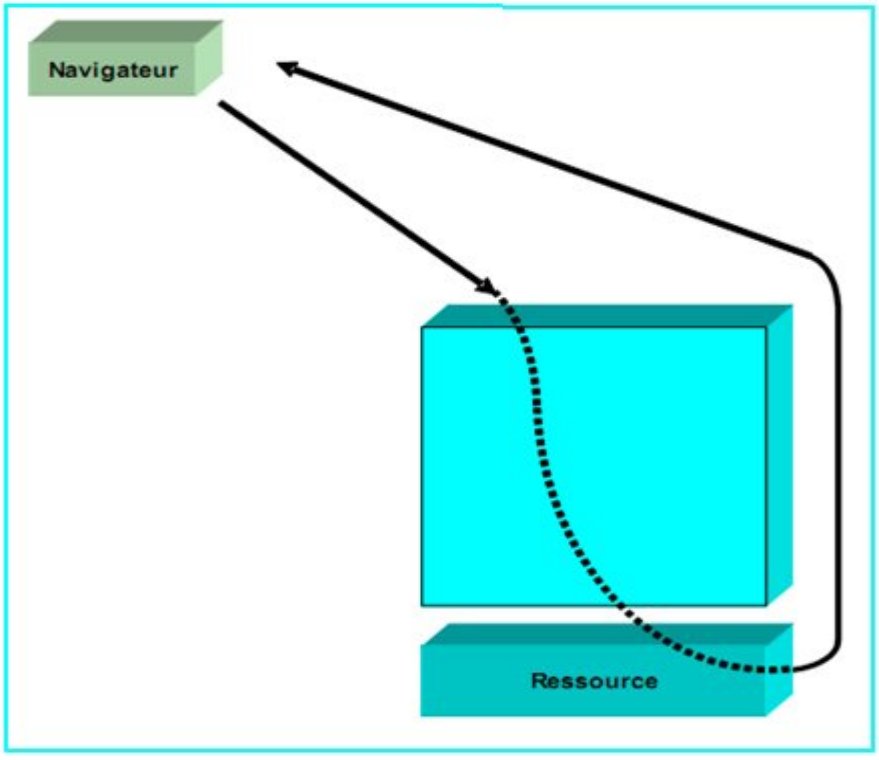


Figure 2.9 – Point de vue de l'utilisateur pour les requêtes suivantes vers un autre SP dans un contexte SSO

2.4.4. Architecture logique du fournisseur de services (SP) :

Le fournisseur de services est composé de trois briques logicielles :

– Le consommateur d'assertions (*Assertion Consumer Service*),

– Le demandeur d'attributs (*Attribute Requester*),

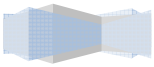
– Le contrôleur d'accès.

Le **consommateur d'assertions** agit comme un pré-filtre. C'est lui qui redirige vers l'IdP lorsque l'utilisateur n'est pas authentifié. Il peut être implémenté au niveau du serveur HTTP (par un module Apache) ou encore par une librairie, appelée par un applicatif web. Lorsque l'utilisateur est authentifié, alors le consommateur d'assertions transmet le *nameIdentifier* au demandeur d'attributs.

Le **demandeur d'attributs** est chargé de la récupération des attributs des utilisateurs auprès de l'IdP. Il peut être implémenté comme un daemon (dédié, interrogeable par les processus du SP) ou par une librairie, interrogeable par un applicatif web. Les attributs récupérés par le demandeur d'attributs sont fournis au contrôleur d'accès.

Le **contrôleur d'accès** est chargé d'autoriser ou non l'accès aux ressources demandées. Il peut être implémenté au niveau du serveur HTTP (par un module Apache par exemple) ou encore par une librairie, appelée par un applicatif web.

La Figure **2.10** montre l'architecture logique d'un SP et son fonctionnement interne lors de la phase d'authentification.



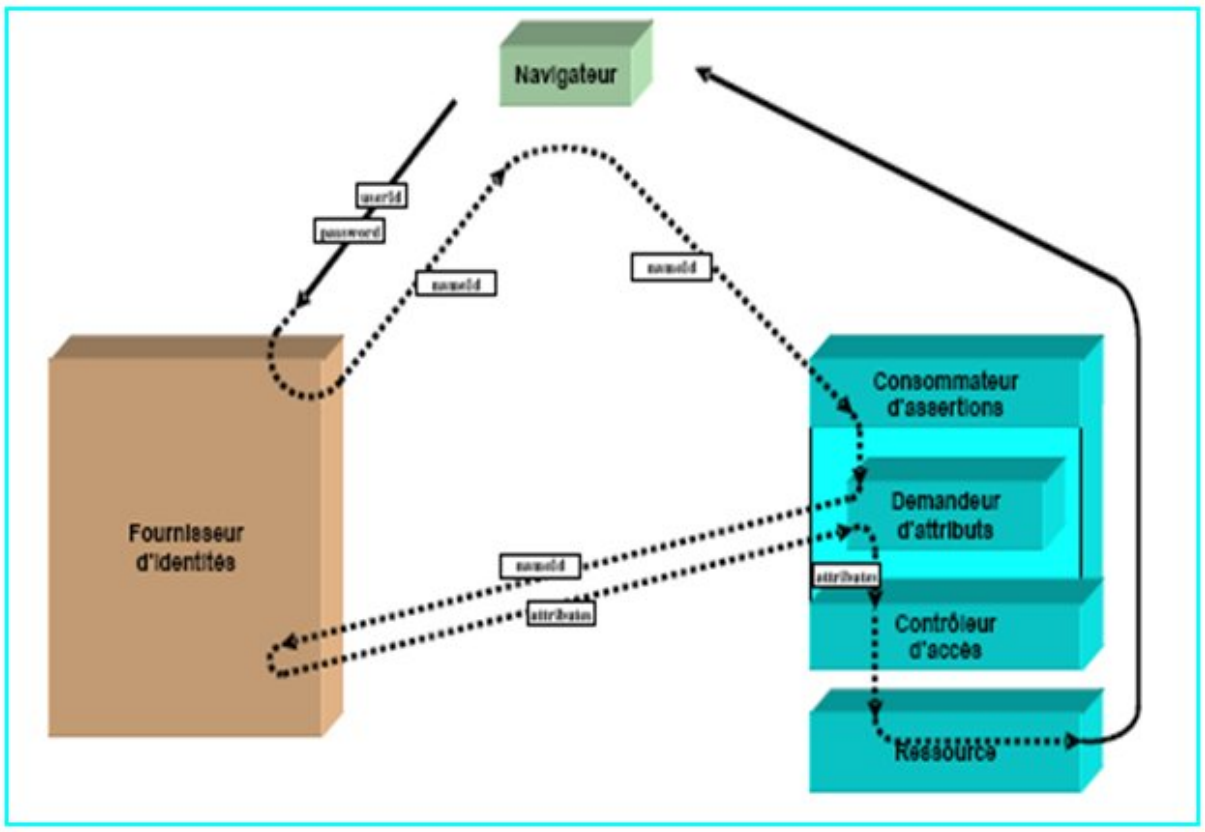


Figure 2.10 – Architecture logique et fonctionnement interne d'un SP

2.4.5. Architecture logique du fournisseur d'identités (IdP) :

Un fournisseur d'identités est composé de trois briques logicielles :

– Le service d'authentification (*Authentication Service*),

– L'autorité d'authentification (*Authentication Authority*),

– L'autorité d'attributs (*Attribute Authrority*).

Le **service d'authentification** est chargé de l'authentification des utilisateurs vis-à-vis de l'ensemble de l'IdP. C'est lui qui, par exemple, demande à l'utilisateur un couple *user*/*password*, puis le valide auprès de la base d'authentification du SI. Les implémentations du service d'authentification peuvent être très variées, depuis un module Apache authentifiant les utilisateurs auprès d'un annuaire LDAP, jusqu'à un client de Single Sign-On comme nous le verrons ultérieurement.

Le service d'authentification n'est pas, si l'on se réfère aux spécifications de Shibboleth [6], partie intégrante de l'IdP ; on ne peut néanmoins pas concevoir d'IdP sans service d'authentification. N'importe quel système d'authentification web est utilisable.

Le service d'authentification est chargé de transmettre à l'autorité d'authentification l'identifiant unique de l'utilisateur au sein du SI (Système d'Information). N'importe quel

système d'authentification web peut être utilisé (formulaire applicatif, royaume HTTP, certificat X509 [5], *Single Sign-On*).

L'**autorité d'authentification** associe le *nameIdentifier* à l'identifiant de l'utilisateur.

L'**autorité d'attributs** délivre, en réponse à une demande d'un SP, les attributs de l'utilisateur correspondant à un *nameIdentifier*, l'association entre l'identifiant de l'utilisateur et le *nameIdentifier* étant maintenue par l'autorité d'authentification. Les attributs de l'utilisateur sont récupérés dans le SI (Système d'Information) de l'établissement, plusieurs sources pouvant être envisagées (annuaire LDAP, base de données…).

La Figure **2.11** montre l'architecture logique d'un IdP et son fonctionnement interne lors de la phase d'authentification.

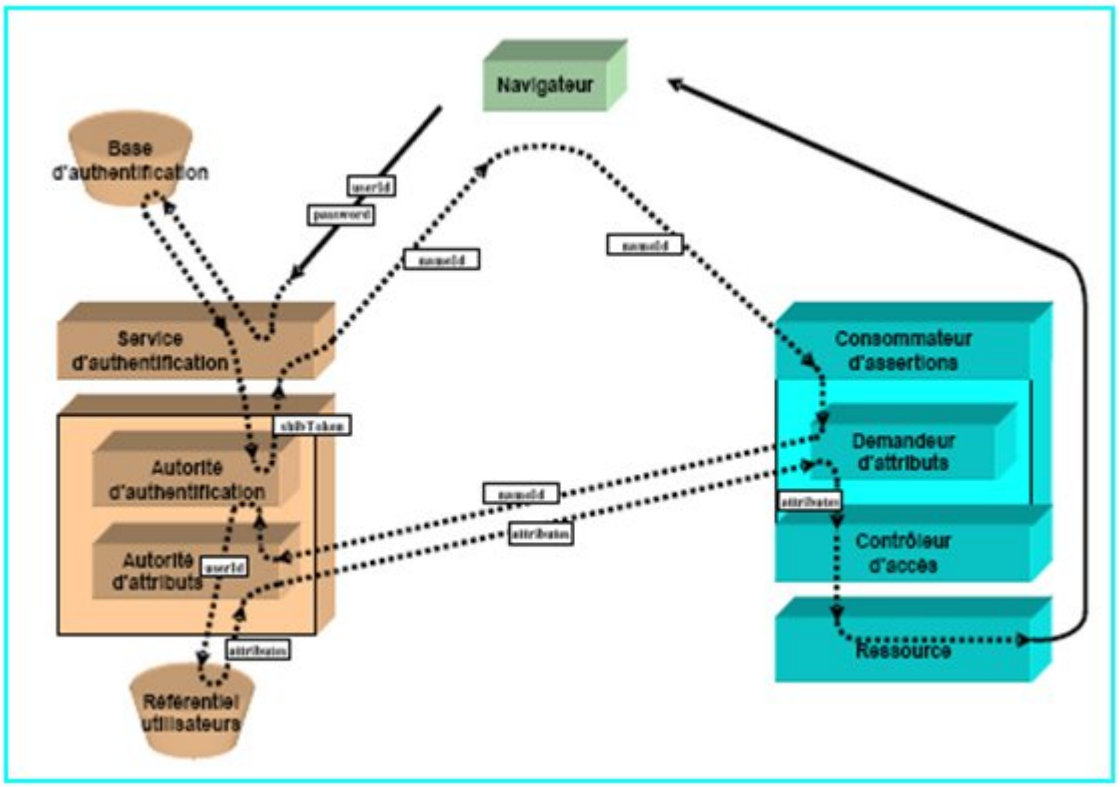


Figure 2.11 – Architecture logique et fonctionnement interne d'un IdP

En résumé, le fonctionnement de shibboleth avec SSO est illustré dans la figure **2.12.**

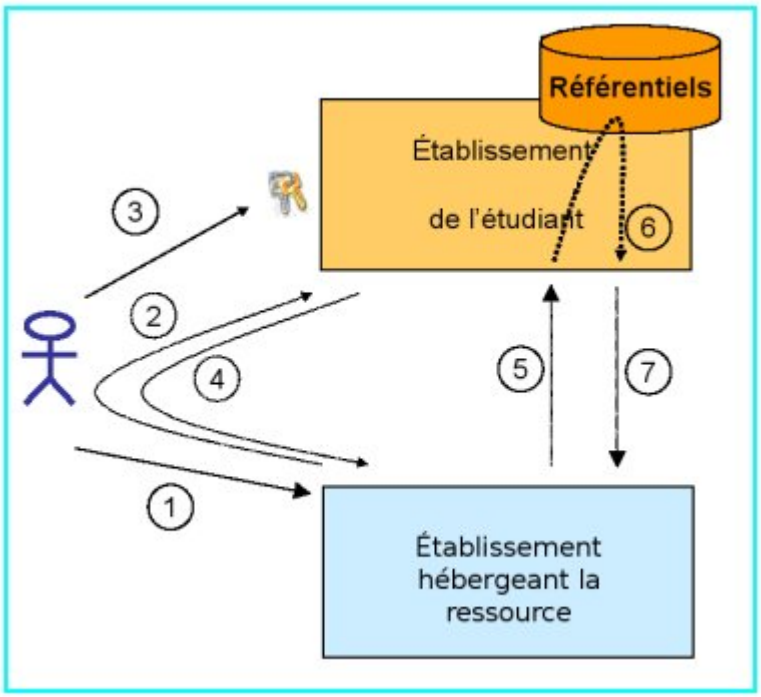


Figure 2.12– Scénario d'authentification d'un Agent

1. Tentative d'accès à la ressource ;

2. Redirection vers le SSO de l'établissement ;

3. Authentification sur le SSO ;

4. Redirection vers la ressource avec une preuve d'authentification ;

5. Demande d'attribut sur l'utilisateur ;

6. Extraction des attributs ;

7. Propagation vers la source ;

2.5. Les méta-données

Les méta-données sont composées d'une liste des membres de la fédération (fournisseurs d'identités et fournisseurs de services) d'une part et d'une liste des autorités de certification de confiance d'autre part. La liste des membres est un fichier XML dont chaque enregistrement fournit les informations requises pour chaque entité à savoir :

– L'identifiant du service, sous la forme d'un URN (*Uniform Resource Name*) ;

– L'intitulé du service ;

– Le contact technique pour le service ;

– L'URL des services correspondants : autorité d'authentification et autorité d'attributs pour un fournisseur d'identités, consommateur d'assertions pour un fournisseur de services.

L'exemple simplifié ci-dessous se limite à la définition d'un fournisseur d'identités et un fournisseur de services.

```
<EntitiesDescriptor
    Name="urn:mace:cru.fr:exemple-federation"
<EntityDescriptor entityID="https://idp.univ-exemple.fr/shibboleth">
    <Organization>
      <OrganizationName xml:lang="fr">Exemple de fournisseur
d'identités</OrganizationName>
      <OrganizationDisplayName xml:lang="fr">Université
d'exemple</OrganizationDisplayName>
      <OrganizationURL xml:lang="en">http://idp.univ-
exemple.fr/</OrganizationURL>
    </Organization>
    <ContactPerson contactType="technical">
      <SurName>Support technique</SurName>
      <EmailAddress>support@idp.univ-exemple.fr</EmailAddress>
    </ContactPerson>
    <SingleSignOnService
    Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
    Location="https://idp.example.org/shibboleth-idp/SSO"/>
    <AttributeService  Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
binding"
    Location="https://idp.univ-exemple.fr:8443/shibboleth-idp/AA"/>
</EntityDescriptor>
<EntityDescriptor entityID="https://sp.example.org/shibboleth">
    <Organization>
      <OrganizationName xml:lang="en">Example Service
Provider</OrganizationName>
      <OrganizationDisplayName xml:lang="en">Services 'R'
Us</OrganizationDisplayName>
      <OrganizationURL
xml:lang="en">http://sp.example.org/</OrganizationURL>
      </Organization>
      <ContactPerson contactType="technical">
        <SurName>Technical Support</SurName>
        <EmailAddress>support@sp.example.org</EmailAddress>
      </ContactPerson>
  <AssertionConsumerService
  index="1"
  isDefault="true"
```

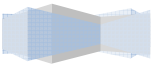
```
  Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-post"
  Location="https://sp.example.org/Shibboleth.sso/SAML/POST"/>
  </EntityDescriptor>
</EntitiesDescriptor>
```

Les méta-données sont gérées de façon centralisée pour la fédération et partagées par tous les sites participants qui doivent mettre en place une tâche périodique de synchronisation. Chaque site peut enrichir ses méta-données s'il participe à plusieurs fédérations ou pour les besoins de relations bilatérales. Ces méta données n'obligent pas un site à avoir des relations avec tous les autres sites qui y sont listés : chaque site est libre de définir avec quels partenaires il travaille.

2.6. Conclusion :

La fédération d'identité sous shibboleth représente une partie de la solution pour la gestion d'identités réparties. Elle est complète, normalisée et disponible en open source.

Donner un sens à cette solution nécessite de l'adosser à des services. Dans notre cas, un de ces services sera représenté par l'accès au réseau WLAN via un portail captif.



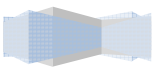
Chapitre 3 :

Le portail captif

3.1. Introduction :

Comme vue précédemment, la fédération d'identité se compose à l'échelle la plus réduite d'un Service Provider (SP) et d'un Identity Provider (IdP). Dans notre cas, le service qui est proposé est le WiFi basé sur un portail captif, alors que pour le fournisseur d'identité, on a fait appel à un annuaire LDAP et un serveur d'authentification RADIUS.

Dans ce qui suit, nous allons présenter les différents modules énumérés ci-dessus.



3.2. Portail Captif : [3]

Si nous souhaitons partager un accès Internet, la principale difficulté réside dans l'authentification. C'est d'autant plus vrai que dans le cadre des réseaux sans-fil, le support reste ouvert. Il existe des solutions liées à la technologie Wifi. Ces solutions, bien que fiables, peuvent être délicates à mettre en place. En effet, l'hétérogénéité du matériel dont disposent les clients ne permet pas toujours le déploiement des dernières technologies. De plus, le nombre potentiel d'utilisateurs peut être très élevé, ce qui interdit les manipulations sur les clients faute de quoi le besoin en support pourrait devenir considérable.

Pour remédier à ce problème sont apparus les portails captifs. Avec ce type de système un utilisateur qui se connecte au réseau pour la première fois est dirigé (capté) vers une page d'authentification. Une fois authentifiée le système autorise l'accès au réseau. Parmi ces logiciels on peut citer NoCat, chiliSpot et monoWall.

Dans notre projet, nous avons développé un portail captif CoovaChilli qui permet d'identifier les utilisateurs.

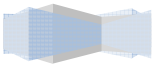
3.2.1. CoovaChilli :

Coovachilli est une application diffusée en open source. Elle permet de contrôler les accès à un réseau sans fil ou filaire via une identification par un portail captif. Coovachilli est entièrement compatible avec le protocole Radius, ce qui permet de déléguer l'authentification, l'autorisation et la journalisation à un serveur Radius tel que Freeradius. Ce programme crée sur la passerelle, un tunnel entre le réseau côté Wi-Fi et le réseau côté internet (le réseau de l'école dans notre cas). Coovachilli laisse entrer dans ce tunnel les paquets des clients dont l'identité a été vérifiée, sinon il y a redirection vers la page d'identification. Coovachilli propose la méthode d'identification suivante :

• Le client sans-fil demande une adresse IP qui lui sera alloué par Coovachilli, qui va faire office de serveur DHCP.

Ensuite quand l'utilisateur veut surfer sur le Web, Coovachilli va intercepter la connexion TCP et rediriger le navigateur vers le serveur WEB permettant l'authentification.

Le serveur WEB va demander à l'utilisateur son login et mot de passe, qui sera crypté dans le cas ou le serveur Web utilise du HTTPS et envoyé au serveur RADIUS qui va regarder si l'utilisateur est connu et autorisé.



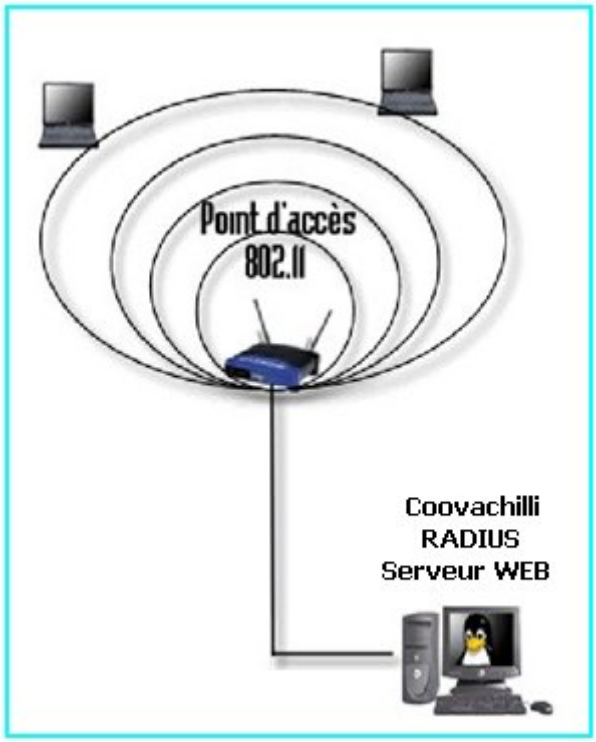


Figure 3.1– Structure d'authentification

3.3. Le Service RADIUS : [10]

3.3.1. Introduction :

Ce protocole ne fait pas partie de la norme 802.11, et il peut être utilisé dans bien d'autres contextes que les réseaux sans fil. Cependant, il est tout à fait central lorsque l'on met en œuvre une architecture sans fil, ce qui est généralement le cas dans un réseau Wi-Fi.

Nous étudierons dans ce qui suit les trois rôles d'un serveur RADIUS : l'authentification des utilisateurs, la définition de leurs autorisations et la comptabilisation de leurs connexions.

3.3.2. Les Fonctions d'un Serveur RADIUS :

3.3.2.1. L'authentification :

a) Un scénario de connexion :

Le ***Remote Authentication Dial In User*** Service (RADIUS) est un protocole défini par l'IETF dans la RFC 2865. De nombreuses RFC viennent étendre ce protocole, qui a été conçu pour être très ouvert. Son nom peut être traduit par « service d'authentification à distance pour des connexions d'utilisateurs ». En d'autres termes, sa fonction première est de centraliser

l'authentification des utilisateurs qui cherchent à se connecter à un réseau ou à un service quelconque. Le scénario élémentaire est le suivant :

Un utilisateur souhaite accéder à un réseau, et pour cela il se connecte à un équipement qui contrôle son accès : cet équipement s'appelle le Network Access Server (NAS), c'est-à-dire le « serveur d'accès au réseau ». Attention : dans le contexte du protocole RADIUS, le NAS est souvent appelé le « client », ce qui peut réellement prêter à confusion.

L'utilisateur fournit son identité au NAS, d'une manière ou d'une autre : le protocole utilisé pour cela n'est pas spécifié par RADIUS ; cela peut être n'importe quel protocole.

En utilisant le protocole RADIUS, le NAS communique alors avec le serveur afin de valider l'identité de l'utilisateur. Si le serveur RADIUS authentifie bien l'utilisateur, il en informe le NAS, et celui-ci laisse désormais l'utilisateur accéder au réseau.

Plusieurs NAS peuvent être configurés pour faire appel au même serveur RADIUS, et lui déléguer le travail d'authentification des utilisateurs. De cette façon, il n'est pas nécessaire à chaque NAS de posséder une copie de la liste des utilisateurs : celle-ci est centralisée par le serveur RADIUS. Dans le cas d'un réseau Wi-Fi, chaque AP peut jouer le rôle de NAS.

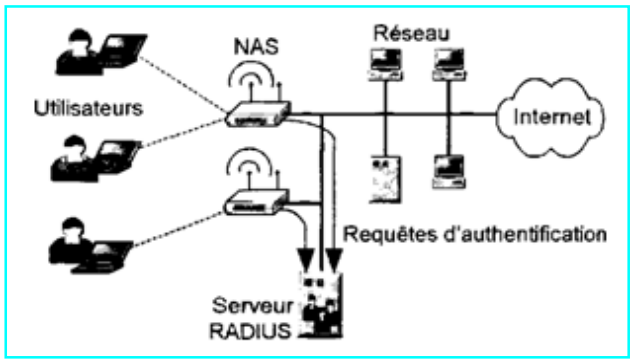


Figure 3.2 – L'architecture RADIUS

Des méthodes d'authentification variées :

Selon les différents serveurs RADIUS qui existent, les méthodes d'authentification prises en charge peuvent varier. Tous sont capables de vérifier l'identité d'un utilisateur grâce à un mot de passe, selon les protocoles PAP ou CHAP, et la grande majorité gère également les protocoles MS-CHAP ou MS-CHAPv2. En outre, la plupart des serveurs RADIUS savent identifier les utilisateurs avec quelques-unes des méthodes EAP, telles qu'EAP/MD5 ou PEAP/MS-CHAP-v2.

La richesse des méthodes d'authentification d'un serveur RADIUS constitue l'un des critères de choix les plus importants.

Les connecteurs :

Pour valider les mots de passe (ou toute autre preuve d'identité), certains serveurs RADIUS consultent simplement un fichier contenant la liste des utilisateurs et leurs mots de passe. D'autres sont capables de lire ces informations dans une base de données relationnelle, comme MySQL ou Oracle. Certains peuvent consulter un serveur LDAP ou un contrôleur de domaine de Windows NT. Certains serveurs RADIUS vous laissent même la possibilité de programmer vous-même votre propre « connecteur» : vous pouvez ainsi relier le serveur RADIUS au système de votre choix, selon la méthode que vous préférez.

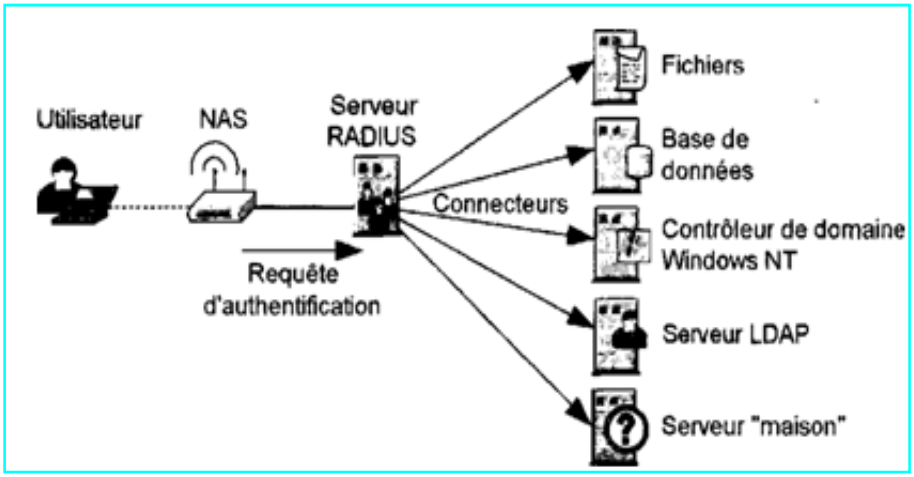


Figure 3.3 – Les connecteurs des serveurs RADIUS

3.3.2.2. L'autorisation :

a) Un paramétrage Fin et dynamique :

Le rôle du protocole RADIUS ne s'arrête pas à la simple authentification. En effet, lorsque le serveur informe le NAS que l'utilisateur est bien authentifié, il peut en profiter pour fournir au NAS toutes sortes de paramètres (on parle plutôt « d'attributs ») utiles pour configurer la connexion de cet utilisateur. Par exemple, il peut indiquer au NAS que cet utilisateur ne doit pas accéder à telle ou telle partie du réseau, qu'il doit être déconnecté au bout de 30 minutes, ou encore qu'il faut lui couper sa connexion s'il télécharge plus de 200 Mo.

Le serveur RADIUS peut finement gérer les autorisations des utilisateurs, en transmettant au NAS des attributs variés. Pour cela, il suffit de configurer le serveur RADIUS en précisant les attributs à renvoyer pour chaque utilisateur ou groupe d'utilisateurs.

b) Les attributs standards :

Quelques attributs possibles pour régler les autorisations des utilisateurs sont définis dans la RFC 2865 : par exemple, l'attribut **Session-Timeout** est défini comme un entier de 32 bits qui représente le nombre de secondes maximum que devra durer la session de l'utilisateur : une fois ce délai écoulé, le NAS doit déconnecter l'utilisateur, de force. L'attribut **Idle-Timeout** est également un entier de 32 bits : son rôle est d'indiquer au NAS au bout de combien de secondes d'inactivité l'utilisateur doit être déconnecté.

3.3.2.3. La comptabilisation :

a) Début de session :

La troisième et dernière fonction d'un serveur RADIUS, définie dans la *RFC 2866*, est de comptabiliser les connexions des utilisateurs. Voici comment cela fonctionne : dès qu'un NAS a reçu du serveur la confirmation de l'authentification d'un utilisateur (accompagnée d'attributs d'autorisation), il envoie une requête au serveur indiquant le début de la session de l'utilisateur. Cette requête comporte de nombreuses informations concernant la session, et notamment :

— L'identifiant de session (Acct-Session-Id) ;

— L'identifiant de l'utilisateur (User-Name) ;

— L'identifiant du NAS (NAS-Identifier) ;

— L'adresse (MAC, en général) de l'utilisateur (Caling-Station-Id) ;

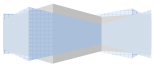
— L'adresse du NAS (Called-Station-Id).

Le serveur enregistre cette information (ainsi que l'heure exacte), dans un simple fichier ou une base de données (ou autres).

b) Fin de session :

Lorsque l'utilisateur met fin à sa session, ou que le NAS le déconnecte (ou encore si la connexion est coupée), le NAS envoie une requête au serveur RADIUS afin de lui indiquer que la session est terminée. Cette requête comporte à nouveau de nombreuses informations au sujet de la session, parmi lesquelles on trouve en général :

— La durée totale de la session, en secondes (Acct-Session-Time) ;



— Le volume total de données téléchargées pendant la session, en nombre d'octets (Acct-Input-Octets) ou en nombre de paquets (Acct-Input-Packets) ;

— Le volume total de données envoyées pendant la session, en nombre d'octets (Acct-Output-Octets) ou en nombre de paquets (Acct-Output-Packets) ;

— La cause de la fin de la session (Acct-Terminate-Cause), par exemple la demande de l'utilisateur (User Request), la perte du signal (Lost Carrier), la fin de la session (Session Timeout) ou encore une inactivité trop longue (idle Timeout) ;

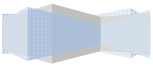
Plus tous les attributs : Acct-Session-Id, User-Name, NAS-Identifier, Calling-Station-Id, Called-Station-Id...

Le NAS peut également être configuré pour envoyer des requêtes à intervalles réguliers, pendant la session de l'utilisateur, afin d'indiquer l'état de la session. Cette requête s'appelle un Intérim-Update, c'est-à-dire une « mise à jour intermédiaire ». Elle peut contenir toutes les informations précédentes.

c) Comptabilisation et administration :

Grâce à la comptabilisation très précise des connexions, il est possible de conserver une trace détaillée de toutes les connexions des utilisateurs. Si l'on possède un bon outil d'analyse des historiques de connexion, il est possible de bien contrôler l'accès au réseau.

On peut ainsi voir, par exemple, quels sont les NAS les plus utilisés, quels sont les utilisateurs qui téléchargent le plus de données, ou encore la durée moyenne d'une session. On peut également détecter des tentatives d'intrusion ou se rendre compte de problèmes de connexion fréquents (Lost Carrier). Bref, c'est un outil très précieux pour l'administrateur réseau.



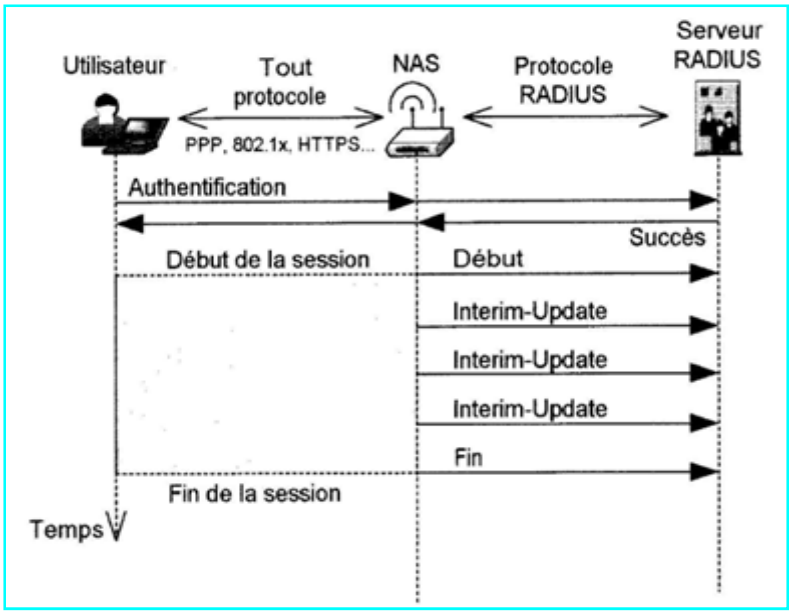


Figure 3.4 – La comptabilisation des connexions

3.3.3. Conclusion :

Le service RADIUS offre donc trois fonctions essentielles : l'authentification des utilisateurs, le paramétrage fin et dynamique de leurs autorisations, et enfin la comptabilisation précise de leurs connexions. Avant de choisir un serveur RADIUS particulier, il faut s'assurer qu'il gère bien les méthodes d'authentification que l'on souhaite mettre en œuvre, et qu'il possède les connecteurs dont on peut avoir besoin, par exemple pour le relier à un contrôleur tel qu'un serveur LDAP.

Le serveur RADIUS nous intéresse particulièrement dans le contexte du Wi-Fi car il est le standard de fait pour le serveur d'authentification.

3.4. LDAP :

3.4.1 Introduction :

Les institutions universitaires ont de plus en plus recours aux réseaux pour accéder à des applications distribuées et à des ressources partagées (sites web, serveurs d'applications, serveurs de fichiers, etc.).

Ces applications et ces ressources doivent interagir avec des ordinateurs situés dans le même réseau local, à travers Internet. Cela nécessite à priori la connaissance des adresses de ces

différentes machines. Or, dans la très grande majorité des cas, on n'utilise jamais les adresses réelles des machines ; on utilise des noms.

Prenons des exemples simples. L'accès à un site web se fera par l'intermédiaire d'un nom désignant le site. L'accès à une imprimante se fera également par l'intermédiaire d'un nom désignant l'imprimante. Ces informations vont être gérées dans une base de données spéciale appelée annuaire. L'annuaire va permettre de transformer le nom du site ou le nom de l'imprimante en une adresse physique permettant aux protocoles de communication d'accéder aux équipements concernés.

3.4.2. Annuaire :

3.4.2.1. Qu'est-ce qu'un annuaire ?

Un annuaire électronique est une base de données spécialisée, dont la fonction première est de retourner un ou plusieurs attributs d'un objet grâce à des fonctions de recherche multicritères. Les objets peuvent être de nature très diverse. Par exemple, un objet de l'annuaire peut représenter une personne et les attributs de cet objet seront alors son nom, son prénom, son numéro de téléphone, etc.

Un annuaire électronique va centraliser des informations et rendre disponibles, via le réseau, à des applications, des systèmes d'exploitation ou des utilisateurs. Il va généralement s'appuyer sur les éléments suivants :

● **Un protocole :** échange des données proprement dit et indication des opérations à effectuer sur ces dernières.

● **Un modèle fonctionnel :** description de la nature des opérations que l'on peut effectuer, comme par exemple une recherche, ou une modification.

● **Un modèle de nommage :** identification des données ; organisation des différentes entrées de l'annuaire.

● **Un modèle d'information :** nature des données pouvant être enregistrées (des chaînes de caractères, des nombres, des numéros de téléphone…).

● **Un modèle de sécurité :** description des services de sécurité permettant d'assurer par exemple le chiffrement des données transférées ou bien l'authentification du client vis-à-vis du serveur.

● **Un modèle de distribution :** création et gestion de serveurs secondaires dans un but de sauvegarde ou de répartition de charge ; création et gestion de liens spéciaux (referrals, méta-annuaires) pointant vers des annuaires responsables d'une partie des données ou vers des annuaires complètement différents.

3.4.3. L'annuaire LDAP :

Cette partie présente de nombreux aspects théoriques d'un annuaire LDAP [11]. Afin de présenter ces différents concepts, le document suivra une progression pas à pas, en allant du plus général au plus détaillé. C'est ainsi par exemple que sera vue tout d'abord la notion d'entrée. Ensuite, la composition d'une entrée sera étudiée, et en particulier la notion d'attribut. Ayant généralement plus d'une entrée dans un annuaire, il conviendra de pouvoir les identifier de manière unique, ce qui amènera au concept de nommage. Puis viendront les notions de classes d'objets permettant de « décrire » la nature intrinsèque de l'objet modélisé par une entrée et finalement celle de schéma regroupant toutes les définitions connues par un serveur.

3.4.3.1. Entrées :

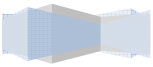
Une entrée dans un annuaire LDAP va permettre de modéliser un objet quelconque du monde réel.

Par exemple, une entrée peut représenter une personne, ou bien une entité administrative d'une université, ou encore une imprimante, un ordinateur, une liste d'utilisateurs, un rôle fonctionnel comme celui d'un directeur, d'une assistante, etc.

Un annuaire va donc se matérialiser sous la forme d'une base de données contenant de nombreuses entrées. Dès maintenant apparaissent de nouveaux besoins : il faut identifier les entrées pour les retrouver ; il faut aussi que chaque entrée puisse contenir différentes informations représentatives de l'objet modélisé par l'entrée. Le premier point va amener le concept de nommage (§ 3.4.3.3) et le second point va impliquer la définition d'attributs (§ 3.4.3.2).

3.4.3.2. Attributs :

Chaque entrée est constituée d'un certain nombre d'attributs. Un attribut permet de représenter un élément distinctif d'un objet du monde réel.



Par exemple, une personne va posséder, entre autres choses, les attributs prénom et nom de famille.

Chaque attribut est défini par un certain nombre de caractéristiques telles que :

— son nom (pour l'identifier facilement) ;

— son identifiant d'objet (pour l'identifier formellement et de manière unique au sein du serveur) ;

— sa syntaxe [11] (pour savoir si l'attribut se comporte par exemple comme une chaîne de caractères, ou bien comme un entier, etc.) ;

— ses règles de comparaison (très utiles pour savoir comment comparer des attributs de même nom dans des entrées différentes).

Ainsi, l'attribut « surname » (nom de famille) possède deux noms pour l'identifier facilement: « surname » et « sn ». Son identifiant d'objet [12], lui, est unique et vaut 2.5.4.4. Un identifiant d'objet est composé d'une suite de nombres entiers permettant de référencer un objet (abstrait ou non) par rapport à une norme, une recommandation, un protocole, etc. La syntaxe de l'attribut « surname » est telle que l'attribut est considéré comme une chaîne de caractères dont la casse[1] n'est pas significative et dont la taille est limitée à 32 768 caractères.

Il est en outre possible de construire de nouveaux attributs en utilisant un mécanisme classique de dérivation : l'**héritage**. Ainsi, par exemple, l'attribut « surname » est en réalité un attribut dérivé de l'attribut « name » ; cet attribut sert également à construire les attributs « commonName », ou « givenName ». Cela signifie que tous ces attributs, bien qu'ayant des noms différents, vont se comporter de la même manière que l'attribut « name ». Cela évite ainsi un grand nombre d'erreurs de définitions d'attributs et simplifie énormément la vie de l'administrateur d'un annuaire LDAP.

3.4.3.3. Nommage :

Le modèle de nommage définit comment sont organisées et identifiées les entrées de l'annuaire. Le plus rapide pour comprendre comment cela fonctionne, est de partir d'un exemple simple.

[1] Casse : (industrie graphique). Pour simplifier, les lettres d'imprimerie sont réparties en deux parties : les lettres capitales (ou haut de casse) et les minuscules (ou bas de casse)

Prenons en effet l'exemple suivant : soit « Utilisateur » travaillant au sein l'ENP. Il a un numéro de téléphone, une adresse électronique, …etc. Toutes ces informations (nom, prénom, identifiant du département, numéro de téléphone, etc.) sont regroupées dans l'une des entrées de l'annuaire (nommée par exemple «cn=mg »). Le département, quant à lui, possède un nom, un numéro de téléphone (celui du secrétariat), un numéro de fax, etc. Ces informations sont également regroupées dans une autre des entrées de l'annuaire (ce sera ici l'entrée dénommée « ou=DEP »). Le département est également dépendant d'une entité administrative plus importante, en l'occurrence l'ENP (entrée « o=ENP»).

Si un utilisateur de l'annuaire souhaite le joindre par téléphone, il lui faudra, par exemple, connaître son nom. L'annuaire devra ensuite accéder à plusieurs entrées, jusqu'à ce qu'il trouve l'entrée contenant son nom, afin de retourner le numéro de téléphone permettant de le joindre.

Pour que l'annuaire puisse accéder à une entrée, il faut :

— un nom pour identifier l'entrée ;

— un chemin pour la trouver dans la base de données.

L'identification d'une entrée se fait à l'aide de ce que l'on appelle le nom distingué (distinguished name ; DN). Ce nom est, par définition et par construction, unique (voir le paragraphe 3.4.3.3.a pour la représentation d'un nom distingué). Il ne peut donc y avoir deux entrées différentes ayant le même nom distingué.

Le chemin pour trouver l'entrée dépend de l'organisation de la base de données matérialisant l'annuaire. Dans le cas de LDAP, la base de données est organisée de manière arborescente. L'arbre, ainsi formé de toutes les entrées, porte le nom de *Directory Information Tree* (DIT). Au sommet de l'arbre se trouve la racine de la base. La racine de la base est également appelée suffixe (cf. § 3.4.3.3.b) ; les deux termes sont équivalents.

Dans le cas de l'exemple, le schéma arborescent des données (cf. Figure**3.5**) indique que l'entrée « cn=mg » est rattachée à l'entrée «ou=DEP » qui est elle-même dépendante de la racine (soit ici, l'entrée « o=ENP»).

Le nom distingué reflète l'organisation arborescente de la base de données. Autrement dit, le nom distingué représente le nom de l'entrée sous la forme du chemin d'accès à celle-ci depuis le sommet de l'arbre, mais présenté de manière inverse.

Dans le cas de l'exemple, le nom distingué correspondant à l'entrée «cn=mg » sera : cn=mg, ou=DEP, o=ENP, c=DZ

L'entrée « ou=DEP » aura pour nom distingué : ou=DEP, o=ENP, c=DZ.

Et enfin, l'entrée « o=ENP», correspondant à la racine de l'arbre proprement dit, aura pour nom distingué : o=ENP, c=DZ.

a) Représentation d'un nom distingué :

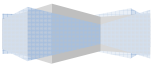
Comme on a pu le deviner dans les exemples précédents, un nom distingué est constitué d'une suite de couples {nomAttribut=valeur Attribut}. Le choix des attributs et des valeurs est a priori libre, à condition que le nom reflète la structure arborescente des entrées.

Autrement dit, dans l'exemple de l'entrée dénommée « ou=DEP, o=ENP, c=DZ », seul le couple initial (ici, « ou=DEP ») est laissé à l'entière liberté de l'utilisateur créateur de l'entrée. Il existe quand même une contrainte : si le choix de cette valeur est en effet libre, il faut simplement qu'elle soit unique à ce niveau de l'arborescence.

La seconde partie (ici, « o=ENP, c=DZ ») est figée ; c'est en réalité le nom de l'entrée située immédiatement au-dessus dans l'arbre des données. Le couple initial porte le nom d'attribut de nommage (naming attribute). Il est souvent exigé que l'attribut de nommage soit également présent dans la liste des attributs d'une entrée, et ce, avec la même valeur.

b) Contextes de nommage :

Toutes les entrées d'un annuaire ont dans leur nom distingué une partie commune : le nom de la première entrée de l'arbre. Le nom distingué de cette entrée initiale est appelée indifféremment « nom de la base », « suffixe de la base » ou plus formellement « contexte de nommage » [13]. Quel que soit le nom sous lequel on la désigne, le nom distingué de cette entrée doit respecter la structure de tout nom distingué, c'est-à-dire une suite de couples {nomAttribut=valeurAttribut}. Toutefois, dans ce cas particulier, le créateur de l'annuaire a une totale liberté sur le choix des attributs et des valeurs. Généralement, le créateur d'un annuaire va choisir comme suffixe pour sa base de données un nom qui reflète la destination de l'annuaire. Mais il est totalement possible de choisir n'importe quel attribut et n'importe quelle valeur pour cet attribut (à condition de respecter la syntaxe de l'attribut).



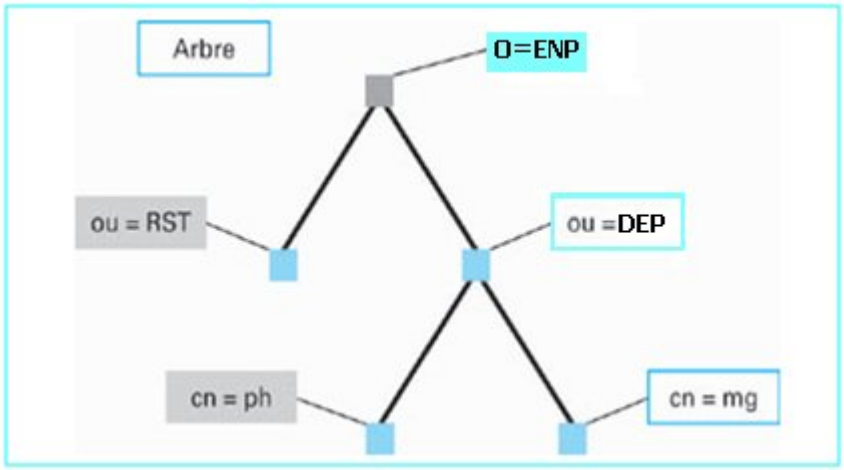


Figure 3.5 – Exemple de schéma arborescent des données

Ainsi, les suffixes suivants sont parfaitement valides, mais parfois « stupides » :

■ cn=toto, cn=titi possibilité d'utiliser plusieurs fois le même attribut (ici, l'attribut«cn» [common Name]), mais avec des valeurs différentes. Dans cet exemple, le suffixe est bien sûr stupide puisqu'il ne donne aucune indication sur la nature de l'annuaire qu'il identifie.

■ ou=secretariat, o=polit-buro, c=su le nombre d'attributs pour un suffixe n'est théoriquement pas limité. Mais en utiliser un trop grand nombre risque vite d'être fastidieux et peut être générateur d'erreurs. Il ne faut pas oublier que lors de l'accès à un serveur, il est souvent exigé que le client doive communiquer le suffixe de la base à laquelle il accède.

■ o=Ecole Nationale Polytechnique, c=DZ la valeur d'un attribut peut être relativement longue et contenir de nombreux caractères diacritiques ou spéciaux. Mais cela est à éviter autant que possible. Certains clients peuvent avoir des difficultés à générer des caractères diacritiques.

■ o=ONU, c=US un exemple de suffixe court mais suffisamment représentatif de l'annuaire géré par le serveur.

3.4.3.4. Classes d'objet :

On peut s'apercevoir par exemple qu'une personne ne possédera jamais l'attribut « adresse IP », et par exemple qu'une imprimante ne possédera jamais l'attribut « nom de famille ». Cela est dû naturellement à la nature intrinsèque des objets modélisés.

Cette nature intrinsèque va être représentée sous la forme d'une classe d'objet. On peut ainsi créer par exemple la classe d'objet « personne » et la classe d'objet « imprimante ».

Le point le plus important est qu'une classe d'objet va permettre de spécifier formellement quels sont les attributs possibles pour une entrée représentant un objet particulier du monde réel.

Cette liste d'attributs est subdivisée en deux parties :

● La première liste définit quels sont les attributs obligatoires que doit posséder une entrée,

● Et la seconde liste définit les attributs optionnels que peut posséder cette entrée.

Ces deux listes imposent des règles strictes sur les attributs que l'on peut utiliser ou non. Prenons un exemple. La classe « person » (les noms de classes de base sont généralement en anglais) permet de modéliser une personne ; toutefois, la classe est simple puisqu'elle ne permet que six attributs au total : deux attributs obligatoires et quatre facultatifs.

Les deux attributs obligatoires sont :

— le nom commun (généralement, pour une personne, c'est le plus souvent l'ensemble prénom – nom) ;

— le nom de famille.

Les attributs optionnels sont :

— le numéro de téléphone ;

— le mot de passe ;

— la description (cet attribut permet d'enregistrer un texte descriptif de l'objet modélisé) ;

— un attribut dénommé « seeAlso » qui est en réalité un pointeur vers une autre entrée dans la base de données.

À l'instar des attributs, les classes d'objet sont définies par plusieurs caractéristiques :

■ En premier lieu, une classe d'objet possède au moins un **nom**, de façon à l'identifier facilement. Elle possède également un **identifiant d'objet** pour l'identifier formellement.

■ Une autre caractéristique d'une classe d'objet est d'indiquer si la classe est **structurelle** ou **auxiliaire**. Les classes structurelles sont les seules qui permettent de construire des entrées. Ce qui signifie que toute entrée doit posséder au moins une classe structurelle. Les classes

auxiliaires permettent de rajouter facilement un certain nombre d'attributs à une entrée existante.

■ Les classes d'objets ont une autre particularité très intéressante : on peut construire de nouvelles classes à partir de classes existantes. Autrement dit, la **notion d'héritage** existe également pour les classes d'objets. Comme on a pu le voir précédemment, le classe « person » est relativement succincte dans sa définition et offre peu de possibilités (par exemple, la classe n'offre pas d'adresse électronique, pas de prénom, etc.). C'est la raison pour laquelle d'autres classes dérivées sont plutôt utilisées, comme par exemple la classe « inetOrgPerson » [14] qui offre une liste impressionnante d'attributs.

Reprenons l'exemple déjà évoqué au paragraphe 3.4.3.3. L'entrée «cn=mg », représentant une personne, possédera la classe « inetOrgPerson ». L'entrée « ou=DEP », représentant une division administrative de l'ENP, possédera, elle, la classe « organizationalUnit ». L'entrée «o=ENP», qui représente le niveau administratif supérieur, pourra être avantageusement décrite grâce à la classe « organization ».

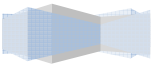
Prenons l'exemple de l'annuaire d'une école. Le personnel de cette école va être représenté sous la forme d'entrées dont la classe sera « inetOrgPerson ». Et ce, que le personnel fasse partie du corps enseignant, ou soit un administratif. Dans le cas du personnel enseignant, il peut être intéressant de rajouter un attribut qui désignera la matière enseignée par cet enseignant. Deux solutions sont possibles : créer une nouvelle classe dérivée de la classe « inetOrgPerson », ou bien créer une classe auxiliaire.

● Dans le premier cas, si on possède déjà une entrée correspondant à un enseignant et que l'on veuille ajouter la matière enseignée, il faudra modifier la classe de cette entrée. Or, ceci ne se fait pas facilement car cela revient à modifier la structure de l'objet.

● Dans le deuxième cas, ajouter l'attribut manquant consistera tout simplement à ajouter la classe auxiliaire ; la classe initiale n'est pas modifiée et la structure de l'objet est inchangée.

3.4.3.5. Schéma :

Un serveur LDAP va gérer plusieurs entrées, chacune comportant une ou plusieurs classes d'objets, comprenant plusieurs attributs, eux-mêmes soumis à des règles syntaxiques, des règles de comparaisons, etc.



L'ensemble des définitions relatives à tous ces objets que sait gérer un serveur LDAP s'appelle le schéma [15] [16]. Le schéma décrit par conséquent les classes d'objets, les types d'attributs et leur syntaxe, ainsi que les règles de comparaison d'attributs connus du serveur.

Le protocole LDAP version 3 exige que le schéma soit publié par un serveur. Celui-ci est enregistré dans des attributs opérationnels.

La publication du schéma est fondamentale. En effet, si n'importe quelle application cliente est capable de reconnaître l'attribut « cn » ou « gn », ou bien la classe « inetOrgPerson », autrement dit les attributs et les classes de bases, il n'en est pas de même pour les attributs et classes définis spécifiquement pour un serveur particulier. La publication du schéma va permettre à ces applications clientes, dans une certaine mesure, de connaître ces nouveaux attributs et ces nouvelles classes. La publication garantit ainsi un plus grand niveau d'interopérabilité.

3.4.4. Le protocole LDAP :

Le protocole LDAP est normalisé par l'IETF [11]. Les données sont échangées dans le format BER [12] (Basic Encoding Rules ; règles d'encodage de base).

Toutefois, le protocole ne respecte pas totalement ce format. En effet, si les valeurs des champs sont bien codés en BER dans le cas de valeurs simples (entiers, booléens, etc.), les autres valeurs sont, en revanche, codées en ASCII, en UTF-8 [17] [18] [19], ou en base 64 [20] selon la nature des données.

Par exemple, les OID (Object Identifier ; cf. [21]) identifiant les attributs ou classes d'objets, ne sont pas transmis selon le codage BER classique des OID. Ils sont transmis sous la forme d'une chaîne ASCII qui est la représentation habituelle « humaine » des OID.

Dans ce qui suit, on verra les différents types de communication entre clients et serveurs (communications client- serveur, et communications serveur-serveur).

3.4.4.1. Communication client-serveur :

C'est le type de communication le plus courant. La communication s'appuie sur le protocole TCP [22], et de manière optionnelle sur TLS [23]. La figure **3.6** donne un exemple simple d'un échange entre un client et un serveur LDAP. Le client ouvre une connexion TCP, se connecte au serveur, émet une requête de recherche (search), et récupère les entrées

correspondantes à sa recherche, puis se déconnecte du serveur (unbind) et ferme la connexion TCP.

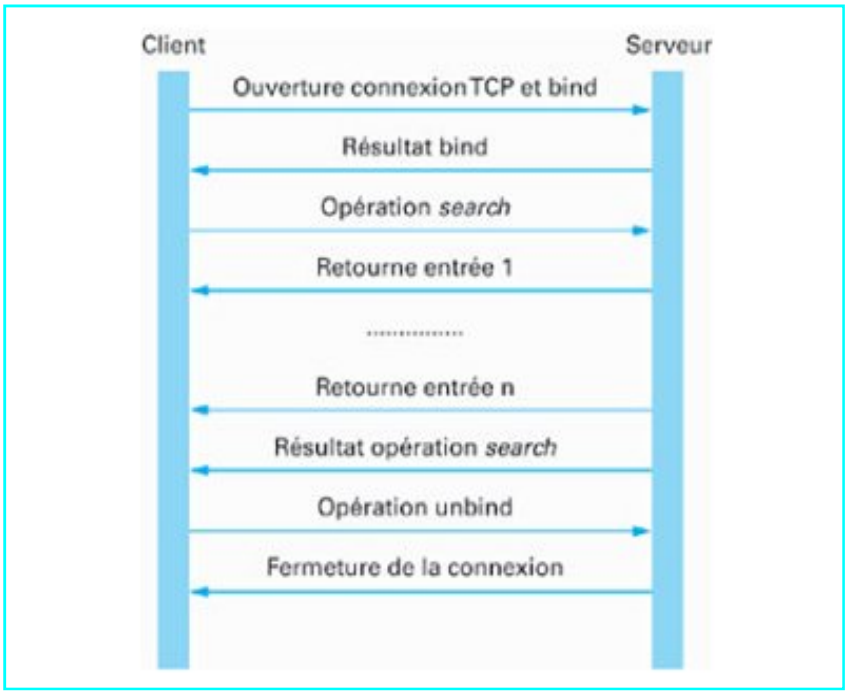


Figure 3.6 – Échange entre un client et un serveur LDAP

3.4.4.2. Communication serveur-serveur :

La communication serveur-serveur est interprétée tout simplement comme un cas particulier de communication client-serveur. L'un des deux serveurs joue alors le rôle du client vis-à-vis de l'autre serveur. Toutefois, on peut définir des services particuliers dans le cadre d'une communication inter-serveurs. Le premier service, de loin le plus utilisé, permet de mettre en œuvre la duplication de serveur (§ 3.4.4.2.a). Un autre service permet de rediriger les requêtes LDAP vers un autre serveur (referral ; § 3.4.4.2.b).

a) Service de duplication :

Le service de duplication permet de disposer d'un serveur maître et d'un nombre indéterminé de serveurs esclaves. Les serveurs sont mis à jour automatiquement en cas de modification du serveur maître.

L'intérêt d'un tel service est d'offrir :

● Une meilleure résistance aux pannes ; si le serveur maître est inactif ou en panne, les serveurs esclaves prennent le relais. Toute- fois, il convient de configurer correctement les

clients pour que ceux-ci puissent rediriger leurs requêtes vers les serveurs esclaves en cas d'indisponibilité du serveur maître.

● La possibilité de gérer facilement plusieurs serveurs miroir.

Un serveur miroir est un serveur LDAP contenant la même information que le serveur maître ; il permet en particulier d'alléger la charge du serveur maître.

Il existe plusieurs protocoles permettant de réaliser cette fonctionnalité. Dans un premier cas, il suffit d'utiliser simplement le protocole LDAP lui-même ; le serveur maître est alors client du serveur esclave, mais il possède des droits d'écriture sur ce dernier. Une autre possibilité consiste à utiliser un autre protocole spécifique permettant de synchroniser le maître et l'esclave (e.g. LDAP synchronization content protocol [24]).

b) Service « referral » :

Ce service permet de relier plusieurs serveurs. Par exemple, le serveur A gère une base de données B_a ; dans cette base, existe une entrée pointant sur une base B_b gérée par un serveur B. Lorsqu'un utilisateur génère une requête vers le serveur A, celui-ci peut rediriger automatiquement la requête vers le serveur B. Ceci peut être particulièrement intéressant lorsque le serveur B gère un sous-arbre du serveur A.

La mise en œuvre de ce service peut poser des problèmes de boucles. En effet, il est possible que le serveur A possède un pointeur vers le serveur B, et que ce dernier possède un lien vers le serveur A. Dans ces conditions, toute requête qui sera transmise par un client au serveur A se verra retransmise au serveur B, puis retournée au serveur A, puis B, et ainsi de suite, formant une boucle infinie.

Aucune solution satisfaisante n'a été trouvée pour résoudre ce problème. Il est simplement demandé aux administrateurs d'annuaires de prendre en considération ce risque et donc de prendre toutes les mesures qui s'imposent pour éviter la formation de telles boucles.

3.4.5. Discussion :

Il est de plus en plus courant de trouver dans la littérature, comme dans des pages web, de nombreuses références à LDAP [26]. Et il semble bien évident que ce phénomène s'accroisse de plus en plus. En effet, il est de plus en plus difficile d'imaginer un monde sans annuaires partagés, et en particulier sans annuaires basés sur LDAP. Ce protocole semble s'être imposé naturellement comme le standard du domaine. Les usages de LDAP couvrent de nombreux

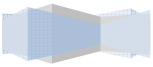
aspects de la vie des établissements, et ce, quelle que soit sa taille. LDAP est en effet utilisé pour la diffusion d'annuaires de type « pages blanches », permet de jouer le rôle de serveur de distribution de certificats, permet l'authentification des utilisateurs, etc.

Cette partie s'attache par conséquent à faire un point sur ce vaste sujet. Il présente en particulier les concepts de base d'un annuaire LDAP : qu'est-ce qu'une entrée, un attribut ? Comment sont identifiées les données au sein de l'annuaire ? Qu'est-ce que le schéma ? etc. Il décrit ensuite la manière d'accéder à un serveur LDAP. Il expose enfin la façon dont sont échangées les données entre un client et un serveur d'annuaire.

Comme cela a été dit plus haut, les annuaires LDAP jouent souvent un rôle non négligeable dans la sécurité des établissements (publication de certificats, authentification des utilisateurs).

3.5. Conclusion:

L'ensemble des briques nécessaires au déploiement de notre service d'accès au réseau Wifi vient d'être présenté. Leurs choix ont été édictés à travers les spécifications qu'ils proposent sans perte de vue de leur adossement impératif à un système de fédération d'identités. L'annuaire reste obligatoire dans le sens où il représente le noyau fondamental d'un système d'information.

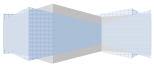


Chapitre 4 :

Développement et déploiement

4.1. Introduction :

Dans ce dernier chapitre, nous allons aborder la partie développement en détaillant les procédures et commandes permettant d'aboutir au résultat escompté, l'objectif étant de mettre en œuvre un service d'accès au réseau sans fil sécurisé sous un fournisseur d'identité Shibboleth.



4.2. Expression et analyse des besoins :

4.2.1. Au niveau des utilisateurs :

4.2.1.1. Connexion au réseau WLAN :

La connexion au réseau wifi doit être simple, c'est-à-dire que l'utilisateur ne doit pas être obligé de saisir une adresse IP pour la machine, la passerelle, le serveur de DNS ou autres paramètres… De plus, le réseau doit être détecté automatiquement par l'ordinateur, l'accès wifi doit s'afficher parmi les réseaux sans fil existants.

4.2.2. Au niveau des administrateurs :

4.2.2.1. Sécuriser l'accès :

Les ondes d'un réseau wifi ne peuvent pas être contenues dans un bâtiment, un ordinateur portable situé en dehors du bâtiment peut capter et se connecter sur le réseau sans fil. Pour éviter ces connexions non autorisées il faut vérifier l'identité de l'ordinateur ou de la personne souhaitant accéder au réseau.

4.3. Mise en œuvre :

4.3.1. Schéma de l'architecture :

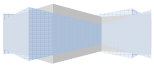
Figure 4.1 – Schéma de l'architecture avec un seul point d'accès

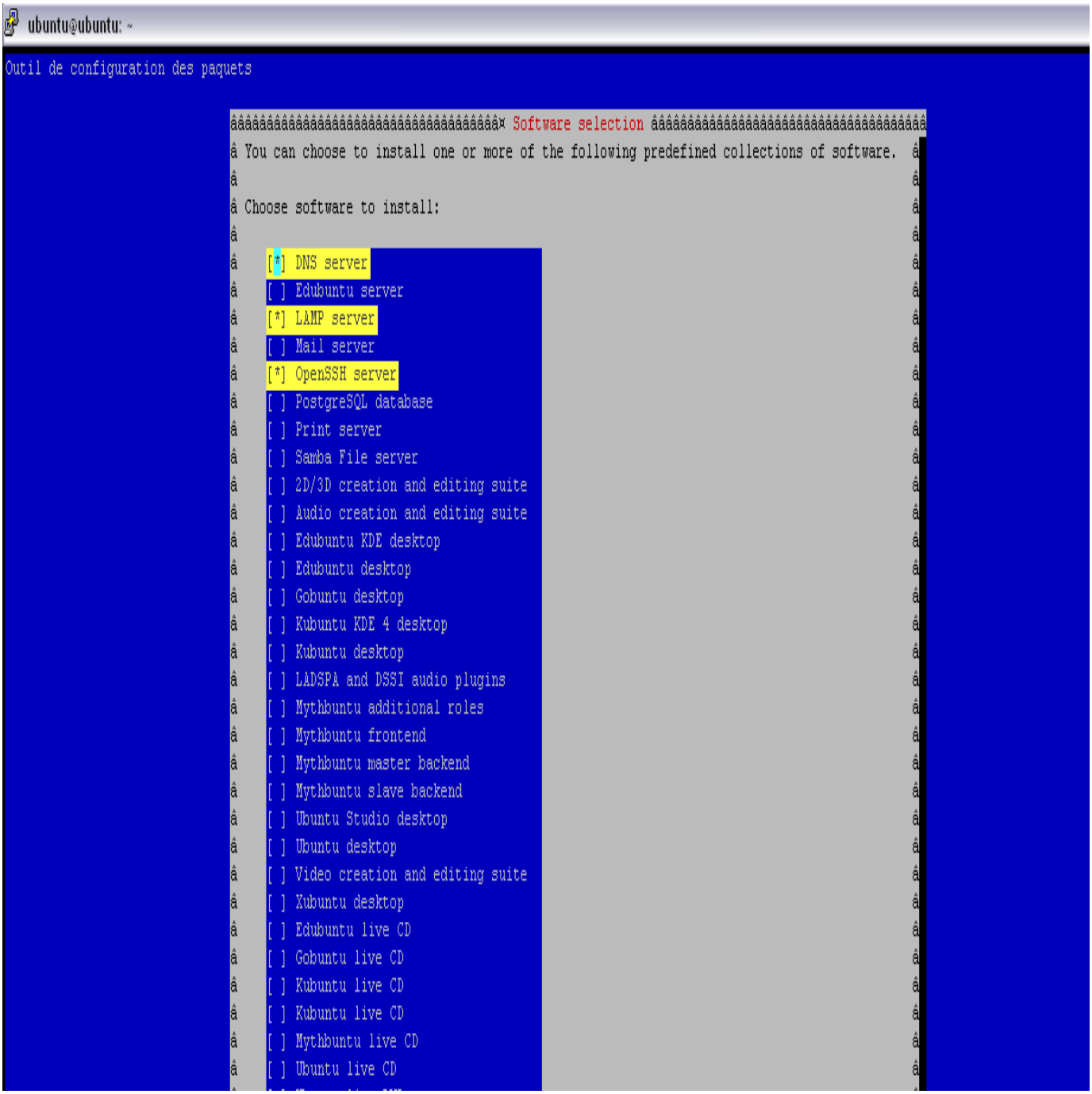
4.3.2. Configuration du serveur :

Cette machine héberge un serveur Radius, un serveur ldap, un annuaire de base de données, un serveur web et Chillispot (CoovaChilli), l'application du portail captif.

4.3.2.1. Installation du serveur :

L'Ubuntu server 8.04 a été installée sur la passerelle Wi-Fi. Cette distribution a été choisie car elle est récente et elle propose les dernières mises à jour logicielles. Nous avons réalisé une installation par défaut, en sélectionnant les services nécessaires (DNS server, LAMP server, OpenSSH server).





4.3.2.2 Configuration réseau :

- Passerelle par défaut : 172.16.13.27
- Serveur DNS : 172.16.13.1
- Nom de machine : passerelle.vdi

a) Interface réseau etho :

Carte réseau connectée au réseau de l'ENP

• Adresse IP : 172.16.13.27

• Masque : 255.255.0.0

b) Interface réseau eth1 :

Carte réseau connectée au point d'accès

• Adresse IP : 192.168.0.253

• Masque : 255.255.255.0

4.3.3. Installation du service Provider (SP):

Cette partie détaille l'installation et la configuration de la brique Shibboleth fournisseur de services (Service Provider, SP). Nous procèderons par étapes pour aboutir à un fournisseur de services Shibboleth opérationnel. La première étape consistera à installer les pré-requis. Ensuite nous installerons la brique logicielle Shibboleth et nous effectuerons une configuration minimale.

4.3.3.1. Pré-requis :

Installation du certificat et de la clé privée associée pour le serveur web :

Un certificat X.509 doit être installé sur le serveur web pour sécuriser les échanges entre le navigateur et le serveur web et permettre une authentification du serveur par le navigateur web.

Si le module mod_ssl pour Apache n'est pas installé, on doit l'installer :

```
# rpm -qi mod_ssl
```

Le paquetage mod_ssl n'est pas installé

```
# yum install mod_ssl
...
Is this ok [y/N]: y
...
Complete!
```

La configuration d'Apache pour utiliser ce certificat se fait comme suit :

```
# vi /etc/httpd/conf.d/ssl.conf
```

```
...
SSLCertificateFile /etc/pki/tls/certs/sp.enp.edu.dz.crt
```

```
...
SSLCertificateKeyFile /etc/pki/tls/private/sp.enp.edu.dz.key
...
SSLCertificateChainFile /etc/pki/tls/certs/AC-test-federation-cachain.pem
```

Configuration du serveur Apache :

Il faut éditer le fichier de configuration de notre serveur Apache pour spécifier les éléments suivants :

• la directive UseCanonicalName doit être positionnée à On ;

• la directive ServerName doit être correctement configurée au nom de votre machine.

```
# vi /etc/httpd/conf/httpd.conf
...
ServerName sp.enp.edu.dz:80
...
UseCanonicalName On
...
```

4.3.3.2. Installation du SP :

```
# cd /usr/local/src
# wget
ftp://ftp.cru.fr/pub/shibboleth/shibboleth/cppsp/latest/RPMS/x86_64/RHE/5/log4shib-
1.0-1.x86_64.rpm
...
```

Renouvelons l'opération pour les 5 autres fichiers à télécharger :

xerces-c-2.8.0-1.x86_64.rpm

xml-security-c-1.4.0-1.x86_64.rpm

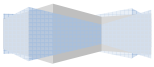
xmltooling-1.1-1.x86_64.rpm

opensaml-2.1-1.x86_64.rpm

shibboleth-2.1-1.x86_64.rpm

Installons les RPM dans cet ordre :

```
# rpm -ivh log4shib-1.0-1.x86_64.rpm
# rpm -ivh xerces-c-2.8.0-1.x86_64.rpm
# rpm -ivh xml-security-c-1.4.0-1.x86_64.rpm
# rpm -ivh xmltooling-1.1-1.x86_64.rpm
# rpm -ivh opensaml-2.1-1.x86_64.rpm
# yum --nogpgcheck install shibboleth-2.1-1.x86_64.rpm
```

L'installation des RPM de la brique SP s'est faite de la façon suivante :

• Les fichiers de configuration installés sont sous /etc/shibboleth/ ;

• le daemon shibd est installé sous /usr/sbin et peut être lancé en utilisant /etc/init.d/shibd start ;

• le module mod_shib et d'autres modules librairies sont installés sous /usr/lib/shibboleth/

•les fichiers de configuration de Shibboleth pointent par défaut vers les emplacements /var/log/httpd/native.log et /var/log/shibboleth/shibd.log pour ce qui est des journaux de log.

4.3.4. Installation de CoovaChilli :

4.3.4.1. Introduction :

Coovachilli est un logiciel open source de contrôleur d'accès, basé sur le projet populaire Chillispot (maintenant défunt), et est maintenu activement par un contributoire Chillispot original.

Coovachilli est un très riche logiciel qui fournit un environnement de portail captif et utilise RADIUS pour l'approvisionnement de l'accès et le « accounting ».

4.3.4.2. Le processus :

Un client qui se connecte à cette interface a tous ses paquets repoussés jusqu'à ce qu'il soit autorisé par la page de connexion pourtant le nom de CoovaChilli (agir comme un suppliant pour certification). Quand un client non-certifié essaie de se connecter à une page web (sur port 80 ou 443) la demande est interceptée par CoovaChilli et est réacheminée à un script perl appelé hotspotlogin.cgi (servi par apache sur http).

hotspotlogin.cgi sert une page à l'utilisateur avec un username et un champ « mot de passe ». Ces données de la certification sont envoyées alors au serveur freeradius qui les fait correspondre à l'information contenue dans la base de données LDAP.

Un utilisateur est alors repoussé ou certifié par freeradius, en incitant hotspotlogin.cgi à présenter un message de refus ou une page avec un message du succès et un lien du logout à l'utilisateur.

4.3.4.3. Installation de CoovaChilli :

Installation :

Pour installer CoovaChilli par paquet, on procède comme suit :

```
wget http://ap.coova.org/chilli/coova-chilli-1.0.13-1 i386.deb
sudo dpkg –i coova-chilli_1.0.13-1_i386.deb
```

Copions les fichiers de configuration par défaut et la configuration d'Apache site:

```
cp /etc/chilli/defaults /etc/chilli/config
mkdir /var/www/hotspot
cd /var/www/hotspot
cp /etc/chilli/www/* /var/www/hotspot
mkdir /var/www/hotspot/uam
cd /var/www/hotspot/uam
wget http://coova.org/uam/
wget http://coova.org/js/chilli.js
```

Modifions index.html et utilisons chilli.js locale :

```
sed -i
's/coova.org\/js\/chilli.js/172.16.13.27\/uam\/chilli.js/g'/var/www/hotspot/uam/index.html
```

Afin de permettre le chargement de coovachilli changeons START_CHILLI à 1, pour cela :

```
vi /etc/default/chilli
START_CHILLI=1
CONFFILE="/etc/chilli.conf"
```

Configuration de base :

Voir **/usr/local/etc/chilli/defaults** pour plus de détails sur les configurations possibles. Copions ce fichier dans config (dans le même répertoire) et éditons-le. Pour charger les réglages et lancer chilli, exécutons la commande « **sudo /usr/local/etc/init.d/chilli start** ». Ceci va générer les fichiers **main.conf, local.conf** et **hs.conf** dans le répertoire « **/usr/local/etc/chilli** ». Afin d'apporter des modifications à la configuration à une date ultérieure, il faut relancer chilli.

Par défaut, il est supposé que Ethernet eth0 est notre connexion à l'Internet et eth1 est l'interface sur la quelle nous voulons avoir des clients (utilisaturs).

```
vi /etc/chilli/config
```

Modifions les lignes suivantes du fichier **config** jusqu'à ce que l'url HS_UAMSERVICE soit définie.

```
# -*- /bin/sh -*-

HS_LANIF=eth1            # Subscriber Interface for client devices
HS_NETWORK=192.168.0.0    # HotSpot Network (must include HS_UAMLISTEN)
HS_NETMASK=255.255.255.0  # HotSpot Network Netmask
HS_UAMLISTEN=192.168.0.253 # HotSpot IP Address (on subscriber network)
HS_UAMPORT=3990          # HotSpot Port (on subscriber network)
```

```
HS_DNS1=172.16.13.1

HS_NASID=nas01
HS_UAMSECRET=uamsecret
HS_RADIUS=127.0.0.1
HS_RADIUS2=127.0.0.1
HS_RADSECRET=radiussecret

HS_UAMALLOW=coova.org, 192.168.0.0/24,www.google.dz


HS_UAMSERVER=192.168.0.253


HS_UAMFORMAT=https://192.168.0.253/cgi-bin/hotspotlogin.cgi

HS_UAMHOMEPAGE=http://\$HS_UAMLISTEN:\$HS_UAMPORT/www/hotspot/coova.html

HS_UAMSERVICE=https://192.168.0.253/cgi-bin/hotspot/hotspotlogin.cgi
```

Installation du pare-feu :

Les développeurs de CoovaChilli ont créé iptables pour définir des règles de filtrage adaptées.

La configuration de CoovaChilli iptables se fait dans le répertoire **/usr/local/etc/piment/up.sh** script. De cette façon, vous savez exactement quelle interface est utilisée par chilli.

```
iptables -I POSTROUTING -t nat -o eth0 -j MASQUERADE
```

Apache server :

La page de login est définie par **hotspotlogin.cgi.** Dans notre cas, le fichier se trouve dans **/usr/share/doc/coova-chilli/hotspotlogin.cgi.gz.**

Nous avons besoin de créer un répertoire dans notre serveur web apache et de copier hotspotlogin.cgi, pour le rendre accessible.

```
sudo mkdir-p / var / www / hotspot / cgi-bin
zcat-c / usr / share / doc / coova-piment / hotspotlogin.cgi.gz | sudo tee / var / www / hotspot / cgi-bin / hotspotlogin.cgi
sudo chmod a + x / var / www / hotspot / cgi-bin / hotspotlogin.cgi
```


Pour terminer, redémarrons notre machine :

```
Reboot
```

Figure 4.2 – Page de redirection

4.3.5. Déploiement des points d'accès :

Pour cette partie, nous commencerons tout d'abord par définir le logiciel utilisé AirMagnet, ensuite, nous donnerons les étapes à suivre pour avoir la couverture et le déploiement optimaux.

AirMagnet Planner est un logiciel qui permet de simuler les points d'accès ainsi que l'antenne et les caractéristiques de l'immeuble avant de prévoir le nombre de points d'accès nécessaires et leur emplacement pour un déploiement WiFi. Ce rapport fournit en temps réel la couverture du signal des points d'accès sur le plan et recommande le nombre de points d'accès nécessaires et leur emplacement sur ce même plan (marqué par des chiffres en rouge).

Ce rapport fournit également des informations détaillées pour les points d'accès en cours de déploiement:

• Nom / adresse MAC du point d'accès

• Channel / SSID attribué

• l'emplacement des coordonnées de l'Access Point

• Hauteur du point d'accès / antenne au-dessus du sol

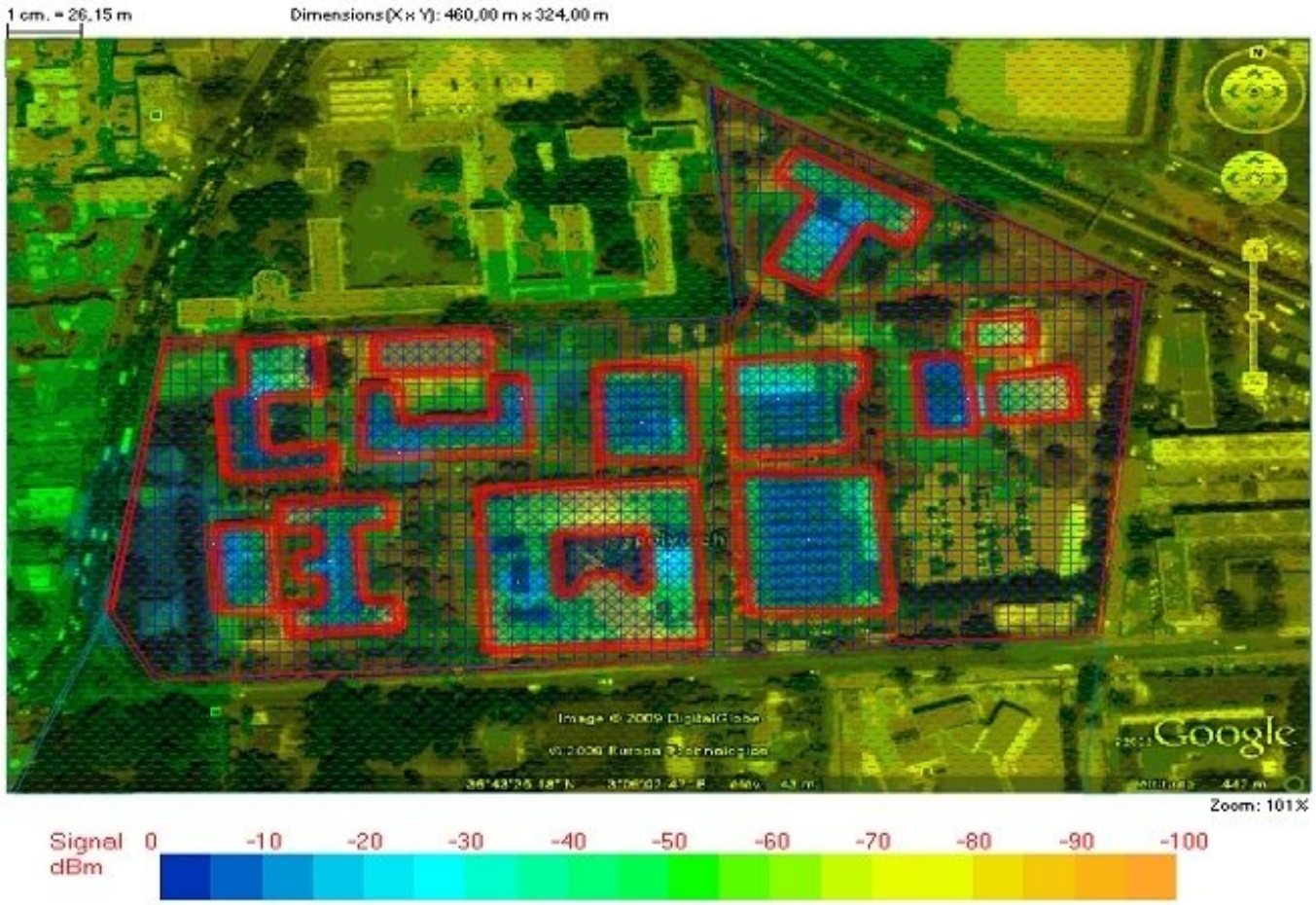
• Type d'antenne et de son cahier des charges

4.3.5.1. Planificateur de couverture du signal

L'image ci-dessous affiche la couverture du signal (en dBm) à chaque point de la carte. Se référer à la légende de la carte ci-dessous pour avoir les valeurs en dBm correspondant à

chaque couleur de la région. En règle générale, les régions avec des signaux en dessous de -67 dBm ne fournissent pas une couverture insuffisante pour un usage normal (cette valeur varie en fonction des besoins des utilisateurs, accords de niveau de service, les applications utilisées, le nombre d'utilisateurs de services, etc.)

Les points d'accès sont affichés dans les lieux où ils sont prévus, ils doivent tenir compte de la puissance et des propriétés de l'antenne. Notez qu'une zone active WiFi peut intégrer une variété de facteurs environnementaux qui peuvent varier tout au long de la journée et affecter la couverture RF projetée.



4.3.5.2. Location Map Planner AP (Access Point) :

L'image ci-dessous affiche la carte du site avec une superposition de grille pour fournir un moyen de décrire l'emplacement de chaque AP (par exemple, un AP dans le coin supérieur gauche de la grille sera décrit par l'emplacement "1-A"). Les AP sont numérotés dans l'ordre qu'ils ont été placés sur le plan. Ces chiffres correspondent à l'AP énumérés à l'AP List (page suivante).

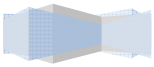
4.3.5.3. Liste Planner AP (Access Point) :

Le tableau ci-dessous répertorie les propriétés de chaque AP sur le plan, y compris son nom, son emplacement (obtenue à partir de la grille à la page précédente), l'adresse MAC, SSID, la taille, le type d'antenne, l'angle d'orientation et le canal.

Name	Location	MAC	SSID	Height
1 AP-BC(BG) **Antenna :Omni-Directional(12dB)**	2-E	00:00:00:00:00:02 **CH:**1	BC **Angle:** 0 **Power:**32(mWatts)	2
2 AP-Annexe.Bibl(BG) **Antenna :Omni-Directional(12dB)**	3-D	00:00:00:00:00:04 **CH:**11	Annexe **Angle:** 0 **Power:**32(mWatts)	2
3 AP-Meta(BG) **Antenna :Omni-Directional(12dB)**	3-D	00:00:00:00:00:06 **CH:**6	Meta SSID **Angle:** 0 **Power:**32(mWatts)	2

4 AP-GI(BG) **Antenna :Omni-Directional(12dB)**	4-D	00:00:00:00:00:08 **CH:**1	GI **Angle:** 0 **Power:**32(mWatts)	2
5 AP-Annexe(BG) **Antenna :Omni-Directional(12dB)**	5-D	00:00:00:00:00:0A **CH:**11	Annexe **Angle:** 0 **Power:**32(mWatts)	2
6 AP-GC(BG) **Antenna :Omni-Directional(12dB)**	5-D	00:00:00:00:00:0C **CH:**1	GC **Angle:** 0 **Power:**32(mWatts)	2
7 AP-SF(BG) **Antenna :Omni-Directional(12dB)**	6-D	00:00:00:00:00:0E **CH:**1	SF **Angle:** 0 **Power:**32(mWatts)	2
8 AP-Chimie(BG) **Antenna :Omni-Directional(12dB)**	6-C	00:00:00:00:00:10 **CH:**11	CHIM **Angle:** 0 **Power:**32(mWatts)	2
9 AP-Meca(BG) **Antenna :Omni-Directional(12dB)**	6-E	00:00:00:00:00:14 **CH:**6	MECA **Angle:** 0 **Power:**32(mWatts)	2
10 AP-GE(BG) **Antenna :Omni-Directional(12dB)**	4-E	00:00:00:00:00:16 **CH:**6	GE **Angle:** 0 **Power:**32(mWatts)	2
11 AP-Hydro(BG) **Antenna :Omni-Directional(12dB)**	3-E	00:00:00:00:00:18 **CH:**1	HYD **Angle:** 0 **Power:**32(mWatts)	2
12 AP-CDC(BG) **Antenna :Omni-Directional(12dB)**	5-E	00:00:00:00:00:19 **CH:**1	CCRSI **Angle:** 0 **Power:**32(mWatts)	2

Tableau 4.1 – Caractéristiques des antennes et points d'accès déployés



4.3.6. Architecture du réseau sans fil :

4.3.6.1. Problématique :

Relier le réseau sans fil au réseau filaire posera problème du partage des utilisateurs des deux réseaux au niveau du serveur d'authentification wifi.

4.3.6.2. Solution :

Afin de rediriger uniquement les utilisateurs du wifi vers le serveur d'authentification wifi, la notion de VLAN doit être introduite.

La solution envisagée est de créer deux VLAN au sein de chaque département, un pour le sans fil et le second pour le réseau local.

Un trunk qui conduira les deux VLAN's (celui du sans fil et du réseau local) sera amené de chaque switch de chaque département vers un switch principal qui regroupera tous les VLAN's.

Par la suite, un deuxième trunk sera crée à la sortie du switch principal pour séparer le VLAN wifi qui sera redirigé vers le serveur d'authentification et le VLAN local qui sera lui redirigé vers la passerelle.

La procédure décrite ci-dessus peut être modélisé à travers l'architecture suivante :

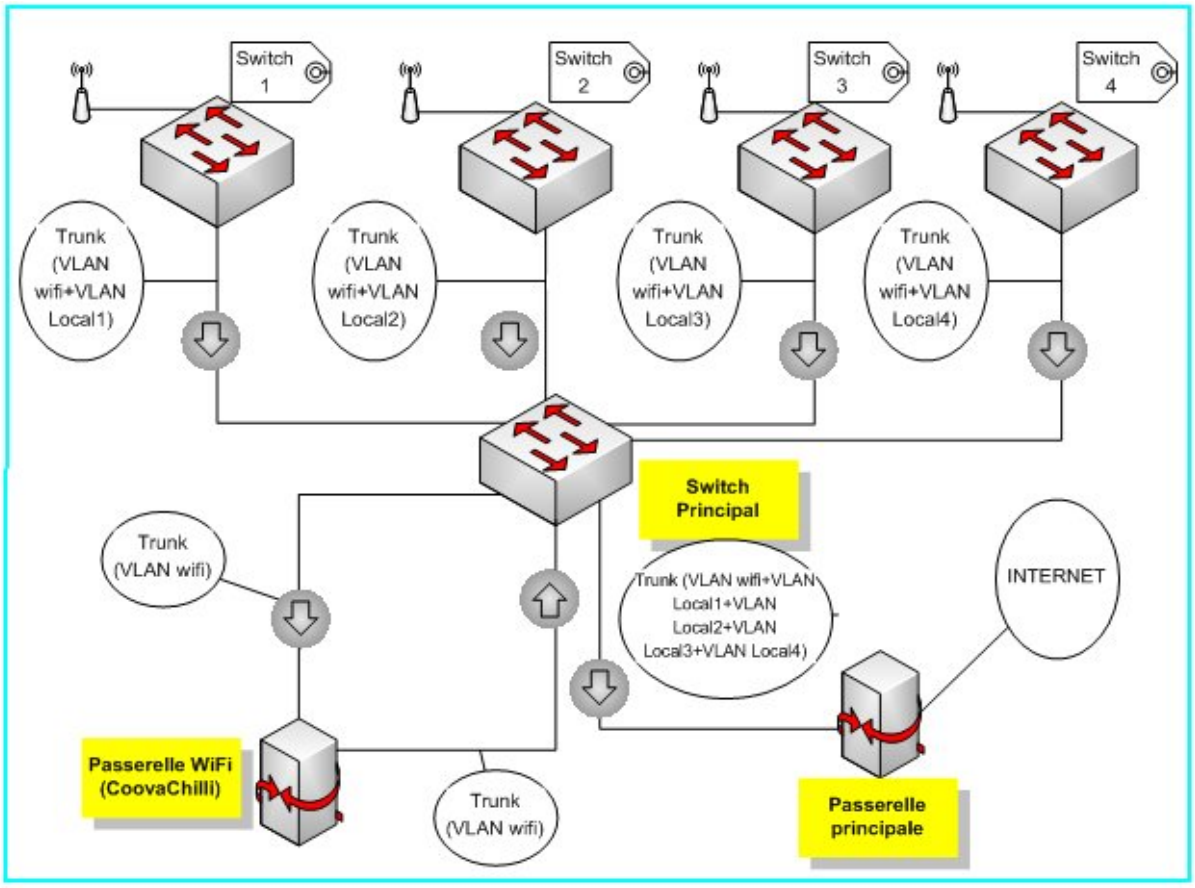


Figure 4.3 – Architecture du réseau sans fil

4.3.7. Installation de la brique fournisseur d'identité :

4.3.7.1. Introduction :

Dans la partie installation de la brique Shibboleth, il est à noter qu'on a utilisé une deuxième machine.

Donc avant l'installation, il faut s'assurer que tous les paquets nécessaires sont installés. Nous aurons besoin de : tomcat, java 1.6, ntp, curl, gnupg, openssl, apache, apache pour mod_ajp.

Pour installer java, nous devons exécuter la commande :

```
sudo apt-get install sun-java6-fonts sun-java-jdk sun-java-jre
```

Pour l'installation de tomcat :

```
sudo apt-get install tomcat5.5
```

Maintenant, plaçons JAVA_HOME dans **/etc/default/tomcat5.5**, il devrait ressembler à :

```
TOMCAT5_USER=tomcat55
```

```
JAVA_HOME=/usr/lib/jvm/java-6-sun
TOMCAT5_SECURITY=no
```

Egalement, plaçons JAVA_HOME dans le fichier **/etc/profile** :

```
JAVA_HOME = / usr/lib/jvm/java-6-sun
 export JAVA_HOME
```

Puis:

```
source / etc / profile
```

La commande ci-dessus permet de faire correspondre JAVA_HOME au lien /usr/lib/jvm/java-6-sun.

4.3.7.2. Installation :

Nous supposons que Shibboleth 2.1.1 a été téléchargé sur notre machine, il est dans le répertoire **/root/workshopefiles**.

Nous devons maintenant décompressez shibboleth identityprovider-2.1.1-bin.tar.gz par :

```
cd / root / workshopfiles
tar zxvf shibboleth-identityprovider-2.1.1-bin.tar.gz tar-zxvf schibboleth
identityprovider-2.1.1-bin.tar.gz
```

Nous devons maintenant aller dans shibboleth-identityprovider-2.1.1 et copier le contenu de **endorsed/** dans **/usr/share/tomcat5.5/common/endorsed/** :

```
cd shibboleth-identityprovider-2.1.1
cp endorsed/* /usr/share/tomcat5.5/common/endorsed/
```

Changeons les permissions du fichier **install.sh** pour qu'il soit exécutable, puis, exécutons-le :

```
chmod + x install.sh
 ./install.sh
```

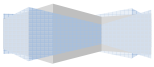
Il nous sera demandé d'entrer un chemin et un nom d'hôte au cours de la procédure d'installation.

Comme chemin, nous donnerons **/opt/shibboleth-idp**, et comme nom d'hôte **idp.enp.edu.dz**.

4.3.7.3. Configuration :

Changeons de propriétaire de **/opt/schibboleth-idp** en tomcat55 et cela comme suit :

```
chown -R tomcat55 /opt/shibboleth-idp
```

À la suite de cette commande, c'est tomcat qui aura tous les droits sur **/opt/schibboleth-idp**.

Plaçons IDP_HOME dans **/etc/profiles** :

```
IDP_HOME=/opt/shibboleth-idp
export IDP_HOME
```

a) Configuration de tomcat:

Nous devons maintenant configurer tomcat pour déployer schibboleth-idp, cela en changeant le répertoire **/usr/share/tomcat5.5/conf/Catalina/localhost** et en créant un contexte descripteur dans le fichier idp.xml. Le fichier idp.xml devrait ressembler à :

```
<Context
    docBase="/opt/shibboleth-idp/war/idp.war"
    privileged="true" antiResourceLocking="false"
    antiJARLocking="false" unpackWAR="true" />
```

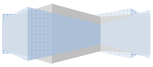
Activons tomcat pour agir en tant que back-end du serveur d'application et acceptant les demandes d'apache, éditons **/usr/share/tomcat5.5/conf/server.xml** et trouvons le fichier de connexion pour le port 8009 :

```
<Connector
    port="8009"
 ...
 />
```

S'il existe, nous devons le remplacer par le code ci-dessous :

```
<Connector
   port="8009" address="127.0.0.1" enableLookups="false"
   redirectPort="8443" protocol="AJP/1.3" tomcatAuthentication="false" />
```

Redémarrons tomcat :



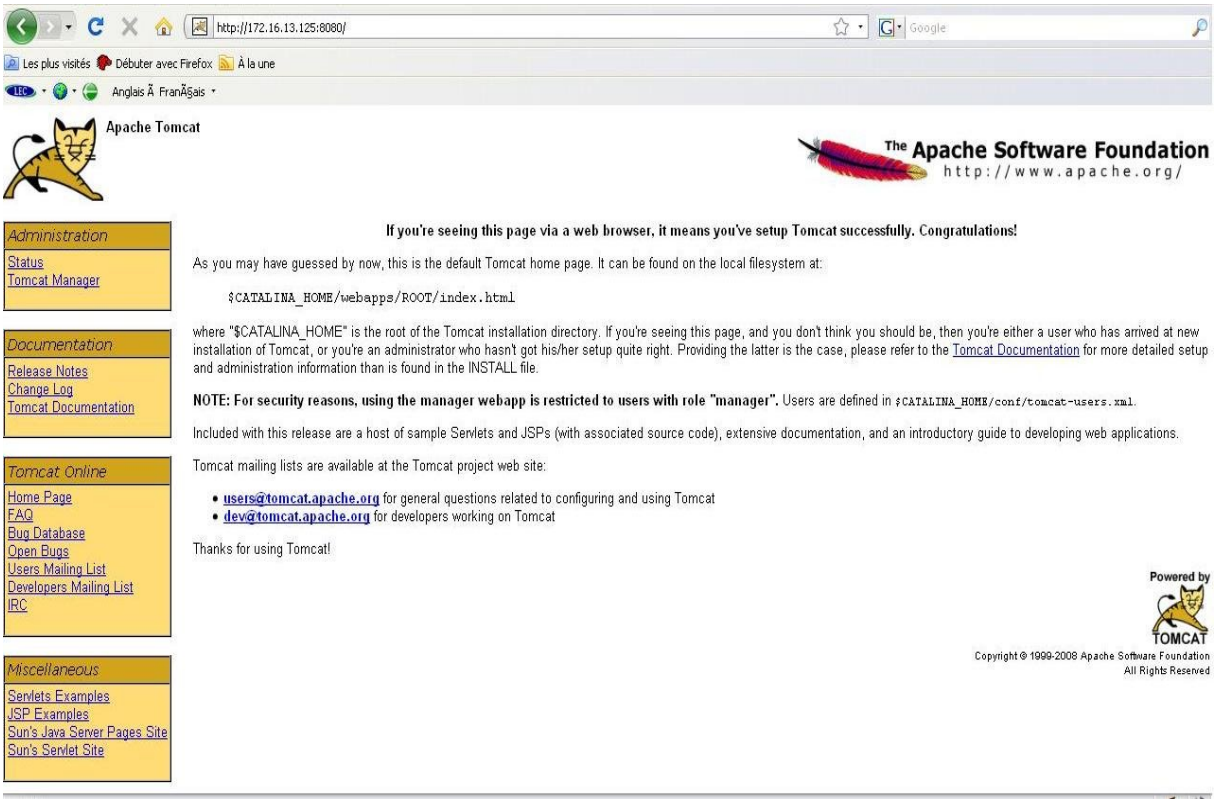


Figure 4.4 – Page principale de tomcat

Shibboleth peut authentifier les utilisateurs dans un certain nombre de moyens, il a sa propre flexible JAAS d'authentification, ou bien il peut dépendre de notre serveur web pour alimenter le « REMOTE_USER » entête http (donc de transmettre l'authentification Apache).

Modifions **/usr/share/tomcat5.5/webapps/idp/WEB-INF/web.xml** et ajoutons /enable :

```
<security-constraint>
   <display-name>Shibboleth IdP</display-name>
   <web-resource-collection>
      <web-resource-name>user authentication</web-resource-name>
      <url-pattern>/Authn/RemoteUser</url-pattern>
      <http-method>GET</http-method>
      <http-method>POST</http-method>
   </web-resource-collection>
</security-constraint>

<login-config>
   <auth-method>FORM</auth-method>
   <realm-name>IdP Password Authentication</realm-name>
   <form-login-config>
      <form-login-page>/login.jsp</form-login-page>
      <form-error-page>/login-error.jsp</form-error-page>
   </form-login-config>
</login-config>
```

Cela configure où shibboleth devrait instaurer un pouvoir, nous allons configurer la manière dont ces pouvoirs sont contrôlés plus tard. Redémarrons tomcat en exécutant :

```
sudo /etc/init.d/tomcat5.5 restart
```

b) Apache 2.2 :

Pour apache, il existe déjà un fichier de configuration de base **/etc/apache2/sites-available/default-ssl.**

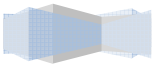
Ajoutons en dessous de la ligne existante, un VirtualHost pour le port 443 (après le <VirtualHost 443> directive)

```
SSLOptions              +StdEnvVars
<IfModule mod_proxy_ajp.c>
   ProxyRequests Off
   <Proxy ajp://localhost:8009>
      Allow from all
   </Proxy>
   ProxyPass /idp ajp://localhost:8009/idp retry=5
</IfModule>
```

Ajoutons un autre VirtualHost (après la clôture VirtualHost tag en bas du fichier(après </VirtualHost>))

```
NameVirtualHost *:8443
<VirtualHost *:8443>
SSLEngine On
SSLCipherSuite
ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:!SSLv2:+EXP
SSLProtocol             all -SSLv2
SSLCertificateFile      /opt/shibboleth-idp/credentials/idp.crt
SSLCertificateKeyFile   /opt/shibboleth-idp/credentials/idp.key
SSLVerifyClient         optional_no_ca
SSLOptions              -StdEnvVars +ExportCertData
<IfModule mod_proxy_ajp.c>
   ProxyRequests Off
   <Proxy ajp://localhost:8009>
      Allow from all
   </Proxy>
   ProxyPass /idp ajp://localhost:8009/idp retry=5
</IfModule>
</VirtualHost>
```

Ajoutons le port 8443 à **/etc/apache2/ports.conf**



```
Listen 80
<IfModule mod_ssl.c>
```

```
    Listen 443
    Listen 8443
</IfModule>
```

Ensuite, permettons ajp_proxy ssl module pour Apache par :

```
sudo a2enmod proxy_ajp
```

Maintenant, redémarrons apache (sudo /etc/init.d/apache2 restart).

c) Configuration de Shibboleth :

Nous avons partiellement configuré shibboleth ci-dessus. Maintenant, nous allons configurer Shibboleth pour vérifier les pouvoirs vis-à-vis ldap qui a été mis en place précédemment.

Les détails du serveur LDAP sont les suivants :

ldap hôte : ldap.enp.edu.dz

ldap port : 389

base dn : ou=people, dc=enp, dc=edu

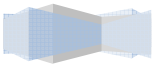
serviceUser : cn=idpconnector, dc=enp, dc=edu

serviceUserPass : secret

domaine de l'utilisateur : cn

Configurons **JAAS** pour ce connecter au serveur LDAP dans le fichier login.config ; Editons /opt/shibboleth-idp/conf/login.config, suprimons le contenu du fichier et remplaçons le par :

```
ShibUserPassAuth {
  edu.vt.middleware.ldap.jaas.LdapLoginModule required
     host="ldap.enp.edu.dz"
     port="389"
     base="dc=enp,dc=edu"
     ssl="false"
     userField="cn"
     serviceUser="cn=idpconnector,dc=enp,dc=edu"
     serviceCredential="secret"
     subtreeSearch="true";
};
```

d) Test:

Pour faire le test, vérifions l'accès à l'adresse: **https://172.16.13.125:8443/idp/profile/Status,** nous obtenons une page avec un « OK », puis testons si sa fonctionne avec apache : **https://172.16.13.125/idp/profile/statut.**

4.3.8. Installation du serveur Radius :

Installation :

```
apt-get install freeradius freeradius-ldap
```

Configuration :

Editons le fichier **/etc/freeradius/radiusd.conf** :

```
sudo vi /etc/freeradius/radiusd.conf
```

Remplaçons le champ par défaut dans le fichier par celui-ci :

```
ldap {
        server = "localhost"
        identity = "cn=admin,dc=enp,dc=edu"
        password = "secret"
        basedn = "dc=enp,dc=edu"
        filter = "(uid=%u)"
        start_tls = no
        access_attr = "uid"
        dictionary_mapping = $(raddbdir}/ldap.attrmap
        ldap_connections_number = 5
        timeout = 4
        timelimit = 3
        net_timeout = 1
}
```

Décommentons la ligne suivante du fichier **/etc/freeradius/radiusd.conf** :

```
# ldap
```

Dans le même fichier **/etc/freeradius/radiusd.conf** et dans la section d'authentification, décommentons les trois lignes suivantes :

```
# Auth-Type LDAP {
#       ldap
# }
```

Editons le fichier **/etc/freeradius/users** :

```
sudo vi /etc/freeradius/users
```

Trouvons les deux lignes suivantes:

```
DEFAULT Auth-Type = System
        Fall-Through = 1
```

Remplaçons les par:

```
DEFAULT Auth-Type = LDAP
        Fall-Through :=1
```

Editons le fichier **/etc/freeradius/clients.conf** :

```
sudo vi /etc/freeradius/clients.conf
```

Décommentons les lignes suivantes :

```
client 127.0.0.1 {
        secret          = testing123
        shortname = localhost
        nastype      = other
}
```

Test :

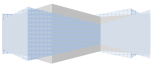
Dans l'exemple suivant, nous allons supposer que nous avons un utilisateur nommé « utilisateur » et son mot de passe est « linux » (sans les guillemets) dans notre répertoire de l'annuaire LDAP

Exécutons la commande suivante:

```
radtest utilisateur linux 127.0.0.1 0 testing123
```

Nous aurons quelque chose comme :

```
Sending Access-Request of id 229 to 127.0.0.1 port 1812
        User-Name = "utilisateur"
        User-Password = "secret"
        NAS-IP-Address = 255.255.255.255
        NAS-Port = 0
rad_recv: Access-Accept packed from host 127.0.0.1:1812, id=229, length=20
```

4.3.9. Installation de l'annuaire LDAP :

Installation

Avant tout, installons le daemon du server ldap (slapd) sur la machine. Pour cela, il suffit d'installer les paquets **slapd ldap-utils**.

```
sudo apt-get install slapd ldap-utils
```

On vous demandera votre mot de passe administrateur et votre nom de domaine. Renseignez-les. Parfois on ne vous demandera que le mot de passe et on ne vous demandera rien concernant le nom de domaine car l'installeur récupère directement le nom de domaine de la machine. Si nous souhaitons renseigner ces champs :

```
sudo dpkg-reconfigure slapd
```

Voici brièvement les réponses attendues pour une installation standard :

```
1.Passer la configuration d'OpenLDAP ? non
2.Nom de domaine ? enp.edu.dz
3.Nom de votre société ? ENP
4.Quelle base de donnée ? hdb
5.Voulez vous que la base de donnée soit effacé lorsque slapd est purgé ? oui
6.Supprimé les anciennes bases de données ? oui
7.Mot de passe administrateur ? ********
8.Confirmez ce mot de passe ? ********
9.Authorisé le protocol LDAPv2 ? non
```

Mais seulement quelques changements seront effectués sur la configuration par défaut. Tout le reste va se jouer dans le fichier **/etc/ldap/slapd.conf**.

Nous allons commencer par enregistrer le mot de passe administrateur (de LDAP) dans le fichier de configuration en éditant ce fichier. Nous allons générer notre mot de passe en chiffré avec la commande :

```
sudo slappasswd
```

On obtient quelque chose dans ce genre :

```
$ sudo slappasswd
New password:
Re-enter password:
{SSHA}d2BamRTgBuhC6SxC0vFGWol31ki8iq5m
```

Cet exemple montre la définition de notre mot de passe en utilisant le mot de passe "secret". (D'après l'implémentation de SSHA, le résultat peut varier) .

Remplir LDAP

L'annuaire a été créé lors de l'installation, il est maintenant temps de le remplir. Il sera rempli avec des entrées classiques qui seront compatibles avec la structure d'un annuaire (pour un annuaire partagé), avec les comptes classiques (pour une authentification Web par exemple). L'annuaire LDAP peut être rempli par des fichiers ldif (ldif signifie ldap directory interchange format). Générons le fichier d'exemple (init.ldif) :

```
$ vi ~/init.ldif
```

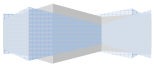
Contenu du fichier :

```
# fichier de données : ~/init.ldif
dn: dc=enp,dc=edu
objectClass: dcObject
objectClass: organizationalUnit
dc: enp
ou: Ecole Nationale Polytechnique

dn: ou=people,dc=enp,dc=edu
objectClass: organizationalUnit
ou: people

dn: ou=groups, dc=enp,dc=edu
objectClass: organizationalUnit
ou: groups

dn: uid=yassine,ou=people,dc=enp,dc=edu
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: yassine
sn: abderrezak
givenName: yassine
cn: yassine abderrezak
displayName: yassine abderrezak
uidNumber: 1000
gidNumber: 10000
gecos: yassine abderrezak
loginShell: /bin/bash
homeDirectory: /home/yassine
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: Yassine.abderrezak@exemple.com
```

```
postalCode: 31000
l: Alger
o: Example
mobile: xx xx xx xx
homePhone:  xx xx xx xx
title: System Administrator
postalAddress:
initials: LP

dn: cn=admin,ou=groups,dc=enp,dc=edu
objectClass: posixGroup
cn: example
gidNumber: 10000
displayName: Example group
```

Dans l'exemple ci dessus, la structure de l'annuaire, c'est à dire un utilisateur et un groupe ont été créés.

Maintenant ajoutons nos entrées à LDAP :

- Arrêt du daemon :

```
sudo /etc/init.d/slapd stop
```

- Suppression ce qui a été ajouté automatiquement à l'installation :

```
sudo rm -rf /var/lib/ldap/*
```

Ajout des données :

```
sudo slapadd -l ~/init.ldif
```

Donnons les droits de lectures aux fichiers de la base de données :

```
sudo chown -R openldap:openldap /var/lib/ldap
```

Relancement de ldap :

```
sudo /etc/init.d/slapd start
```

Nous allons pouvoir vérifier que les données ont été correctement ajoutées avec les outils du paquet ldap-utils.

Pour effectuer une recherche dans les annuaires LDAP il nous suffit de faire :


```
ldapsearch -xLLL -b "dc=enp,dc=edu" uid=yassine sn givenName cn
dn: uid=yassine,ou=people,dc=enp,dc=edu
cn: yassine abderrezak
```

```
sn: abderrezak
givenName: yassine
```

Une rapide explication :

- -x désactive l'authentification SASL ;
- -LLL empêche l'affichage des informations LDIF ;
- -b indique la branche utilisée.

Utilisation du serveur LDAP :

Maintenant que notre serveur est prêt et démarré nous pouvons :

- Authentifier nos utilisateurs dans l'annuaire. LDAPClientAuthentification
- Authentifier vos utilisateurs via une application web
- Utiliser l'annuaire comme une base de données pour votre client mail

phpLDAPadmin :

Nous devons installer les paquets **php5-ldap phpldapadmin**.

```
Sudo apt-get install php5-ldap phpldapadmin
```

Et on y accède via : http://172.16.13.27/phpldapadmin/

Attention, le login est : "cn=admin,dc=enp,dc=edu".

Lorsque nous allons sur l'interface web de phpldapadmin, nous avons cette erreur :

Memory Limit low. Your php memory limit is low - currently 16M

Pour palier à ce problème nous avons édité en admin le fichier php.ini :

```
cd /etc/php5/apache2/
 sudo vi php.ini
```

Nous avons trouvé la section suivante et nous avons changé la valeur memory_limit = 16M en mettant 64M :

```
;;;;;;;;;;;;;;;;;;;
 ; Resource Limits ;
 ;;;;;;;;;;;;;;;;;;;
 max_execution_time = 30     ; Maximum execution time of each script, in seconds
 max_input_time = 60 ; Maximum amount of time each script may spend parsing
request data
```

```
;max_input_nesting_level = 64 ; Maximum input variable nesting level
memory_limit = 64M      ; Maximum amount of memory a script may consume (16MB)
```

Il suffit ensuite de recharger la configuration d'apache pour que la modification soit prise en compte :

```
sudo /etc/init.d/apache2 reload
```

Puis relançons notre serveur ldap :

```
sudo /etc/init.d/slapd restart
```

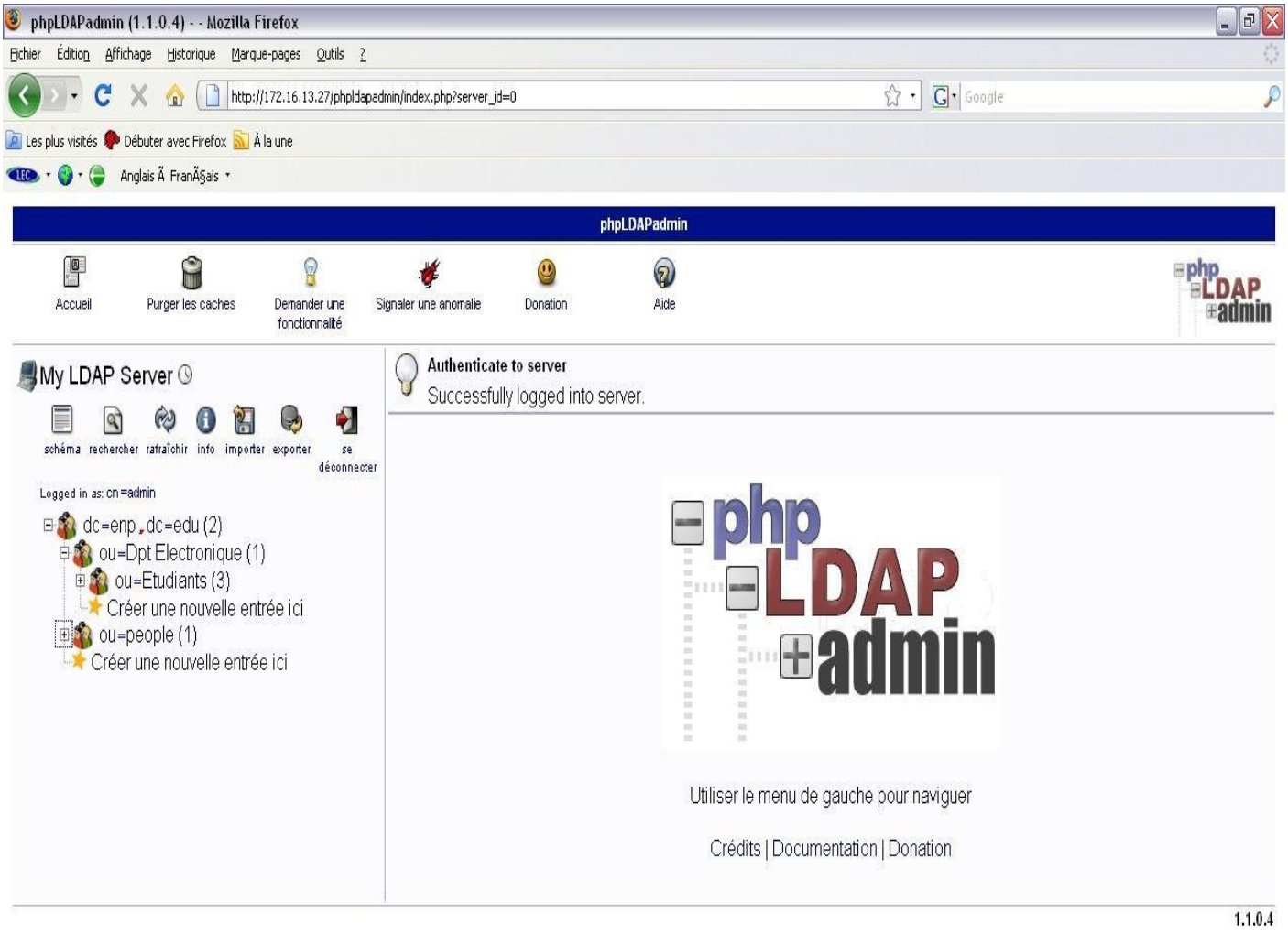
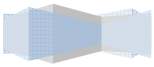


Figure 4.5 – phpLDAPadmin

4.4. Conclusion

La finalité de ce chapitre a répondu aux deux objectifs naturels du déploiement d'un service donné à travers une fédération d'identité

C'est ainsi que le réseau Wifi a était déployé tenant compte des contraintes d'espace mais aussi de son adéquation avec le réseau logique existant, c'est ainsi que les Vlan's y ont été introduit sachant les problèmes de sécurité qui peuvent être posé. Ce service a été couplé avec un service d'identification adossé à Shibboleth, outil de fédération d'identité que nous avons déployé.



Conclusion Générale :

Après trois mois d'effort, nous avons pu atteindre notre objectif, qui est le déploiement d'un réseau sans fil WiFi sécurisé sous un fournisseur d'identité Shibboleth.

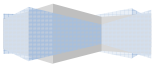
L'objectif de cette étude est d'étendre le réseau filaire pour donner aux utilisateurs une mobilité accrue, et une liberté d'utilisation du réseau tout en garantissant une meilleure sécurité en donnant la possibilité d'intégrer une fédération d'identité.

Nous avons présenté dans ce mémoire une solution complète pour déployer un réseau sans fil dans n'importe quel établissement.

Nous avons tout d'abord effectué une étude du plan de l'école pour pouvoir déployer de manière convenable et optimale les différents points d'accès WiFi.

Ensuite, nous avons abordé la partie configuration, qui consiste à installer les différents serveurs pour pouvoir authentifier et autoriser l'accès au service WiFi.

Ce travail pourra être amélioré en abordant la notion de sécurité au niveau de la couche physique et cela en intégrant les protocoles dédiés à cet effet.



Glossaire :

802.1x — Norme de l IEEE pour le contrôle d'accès à un réseau. Le contrôle est exercé au niveau d un port d'un commutateur, ou pour chaque association dans un AP. Ce standard repose sur l'EAP, et l'authentification des utilisateurs est généralement réalisée par un serveur RADIUS.

802.11 — Norme conçue par l'IEEE en 1997 pour les réseaux locaux sans fil, et constamment améliorée depuis. Elle d finit trois couches physiques (infrarouge, FHSS et DSSS sur les fréquences de 2,4 GHz) et une couche MAC offrant de nombreuses fonctionnalités : partage du média, fragmentation, économie d'énergie, sécurité.

802.11a — Amélioration du 802.11 sur les fréquences de 5 GHz. Grâce à la modulation radio OFDM, cette variante du Wi-Fi peut atteindre un débit théorique de 54 Mb/s.

802.11b — Amélioration du 802.11 DSSS, cette variante du Wi-Fi peut atteindre un débit théorique de 11 Mb/s grâce à la modulation radio HR-DSSS.

802.11c — Précisions destinées aux constructeurs d'AP (pour le mode bridge).

802.11d — Précisions pour les constructeurs de matériel Wi-Fi (internationalisation).

802.11e — Amélioration de la couche MAC du 802.11, destinée à permettre une meilleure gestion de la QoS.

802.11f — Définit l'IAPP (Inter-Access point roaming protocol).

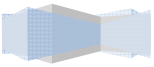
802.11g — Amélioration du 802.11b : elle peut atteindre 54 Mb/s grâce à la modulation radio OFDM.

802.11h — Adaptation du 802.11a à la législation européenne.

802.11i — Nouvelle norme de sécurité pour le 802.11, en remplacement du WEP.

802.11j — Adaptation du 802.11 à la législation japonaise.

802.11k — Amélioration du 802.11 pour faciliter les mesures radio.



802.11m — Groupe de travail du 802.11 chargé de la maintenance de la norme.

802.11n — Amélioration du 802.11a destinée à dépasser 100 à 500 Mb/s en 2006.

802.2 — Norme de l'IEEE définissant la couche réseau LLC.

A

AAA — Un serveur AAA (Autorisation, Authentification, Accounting) gère l'authentification des utilisateurs, leurs autorisations et la comptabilisation de leurs connexions (voir aussi RADIUS).

Ad Hoc — Dans un réseau Wi-Fi de type Ad Hoc, les stations communiquent directement entre elles plutôt que par le biais d un AP (voir aussi Infrastructure).

AP — Access Point (point d'accès) : borne Wi-Fi composant l'ossature d'un réseau sans fil. En mode Infrastructure, tout utilisateur doit passer par un AP pour accéder au réseau sans fil : tout son trafic est alors relayé par l'AP auquel il est « associé ».

ASCII — American Standard Code for Information Interchange « Code américain normalisé pour l'échange d'information ») est la norme de codage de caractères en informatique la plus connue et la plus largement compatible. C'est également la variante américaine du codage de caractères ISO/CEI 646. ASCII contient les caractères nécessaires pour écrire en anglais. Elle a été inventée par l'américain Bob Bemer en 1961. Elle est à la base de nombreuses autres normes comme ISO 8859-1 et Unicode qui l'étendent.

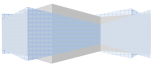
ASN.1 — Abstract Syntax Notation One est un standard international destiné à l'origine à décrire les données échangées dans les protocoles de télécommunication (modèle OSI).

B

BER — Le codage **Basic Encoding Rules** (règles d'encodage basiques) est un des formats d'encodage défini par le standard ASN.1.

BSS — Basic Service Set. Un réseau Wi-Fi composé d'un seul AP.

BSSID — Identifiant d'un BSS. Il s'agit d'un nombre de 48 bits, égale à l'adresse MAC de l AP en mode Infrastructure, ou aléatoire en mode Ad Hoc.



C

CHAP — Challenge Handshake Authentication Protocol est un protocole d'authentification pour PPP à base de challenge, ce qui le rend bien plus sûr que son pendant PAP. L'objectif de CHAP est que le pair s'authentifie auprès d'un authentificateur sans échange de mot de passe en clair sur le réseau et sans que l'échange puisse être rejoué par un tiers à l'écoute. La contrainte est que chaque partie partage un « secret » (mot de passe) commun. Microsoft a développé la variante MS-CHAP qui supprime cette contrainte.

D

DHCP — Dynamic Host Configuration Protocol (DHCP) est un terme anglais désignant un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une station, notamment en lui assignant automatiquement une adresse IP et un masque de sous-réseau.

DN — Distinguished Name d'un objet est un moyen d'identifier de façon unique un objet dans la hiérarchie. Un DN se construit en prenant le nom relatif de l'élément (RDN - Relative Distinguished Name), et en lui ajoutant l'ensemble des noms relatifs des entrées parentes. Le DN d'un élément est donc la concaténation de l'ensemble des RDN de ses ascendants hiérarchiques.

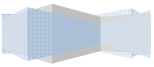
DNS — Domain Name System (ou DNS, système de noms de domaine) est un service permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine.

DS — Distribution System. Il s'agit du lien entre les AP d'un réseau Wi-Fi de type Infrastructure. Généralement le DS est le réseau filaire auquel sont reliés les AP, mais il peut également s'agir d'un lien sans fil.

DSSS — Direct Sequence Spread Spectrum. Modulation radio utilisée par le 802.11b et le 802.11g.

E

EAP — Extensible Authentication Protocol est un mécanisme d'identification universel, fréquemment utilisé dans les réseaux sans fil tel que Wi-Fi.



ESS — Extended Service Set. Réseau Wi-Fi de type Infrastructure, pouvant être composé de plusieurs BSS.

ESSID — Identifiant d'un ESS, souvent noté simplement « SSID ». Il s agit d'un nom composé au maximum de 32 caractères.

ETSI — European Telecommunications Standards Institute. Institut européen des normes de télécommunication, similaire à l'IEEE.

F

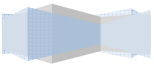
Fédération— ensemble d'organisations dont les membres coopèrent pour partager des services. Les différents acteurs d'une fédération sont les fournisseurs d'identités, les fournisseurs de services et le service de découverte. Le contrôle d'accès à un service est maîtrisé par l'organisation propriétaire du service, mais l'authentification des utilisateurs est déléguée à l'organisation à laquelle appartient l'utilisateur. Cette dernière peut également propager des attributs des utilisateurs jusqu'aux services selon leurs besoins. Ces mécanismes affranchissent les fournisseurs de services de la gestion en local des utilisateurs et de leur authentification. Quand aux utilisateurs, ils évitent l'apprentissage d'un nouveau mot de passe pour chaque service.

FHSS — Frequency Hopping Spread Spectrum. Modulation radio qui consiste à sauter régulièrement d'un canal d'émission à un autre. Cette technique permet de mieux résister aux interférences localisées dans le spectre. Elle a été plus ou moins abandonnée par le Wi-Fi, mais est à la base du Bluetooth et du HomeRF.

H

HTTP — Le HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

Hotspot — Zone d'accès à l'Internet par le Wi-Fi, en général payant.



I

IdP — fournisseur d'identités (ou *Identity Provider*, IdP) : organisation membre d'une fédération et qui gère l'identité informatique d'un ensemble d'utilisateurs : création, suppression et maintenance de leurs informations d'identification (par exemple une universités avec ses étudiants et ses personnels). Un fournisseur d'identités offre un service d'authentification à ses utilisateurs, qui leur permet de s'authentifier sur le réseau. Lorsqu'un utilisateur veut accéder à un service offert au sein de la fédération, il utilise le service d'authentification de son organisation d'appartenance pour s'authentifier. Un fournisseur d'identités peut définir les attributs de ses utilisateus qu'il s'autorise à propager aux fournisseurs de services.

IBSS — Independent BSS. Réseau composé de plusieurs stations en mode Ad Hoc.

IEEE — Institute of Electrical and Electronics Engineers. Organisme de standardisation américain, notamment à l'origine du 802.11, sur lequel le Wi-Fi repose.

IETF — Internet Engineering Task Force. Organisme informel à l'origine des principaux standards de l'Internet.

Infrastructure — Dans un réseau Wi-Fi de type Infrastructure, chaque station est associée un AP et ne communique que par son intermédiaire (voir aussi Ad Hoc).

ISO — International Organization for Standardization.

L

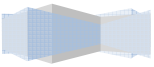
LAMP — est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites Web. L'acronyme original se réfère aux logiciels suivants :

- « **L**inux », le système d'exploitation ;

- « **A**pache », le serveur Web ;

- « **M**ySQL », le serveur de base de données ;

- « **P**HP » à l'origine, puis « **P**erl » ou « **P**ython », les langages de script.



LAN — Local Area Network. Réseau de dimension « locale » : réseau d'entreprise, réseau familial, etc.

LDAP — Lightweight Directory Access Protocol. Protocole d'accès à un annuaire.

LDIF —LDAP Data Interchange Format est un format standardisé d'échange de données, qui permet la représentation des données contenues dans un annuaire LDAP.

LLC — Logical Link Control. Couche réseau définie par l IEEE (802.2), au-dessus de la couche MAC. Elle sert d'interface unique entre les couches 2 et 3 du modèle OSI.

M

MAC — Media Access Control. Couche réseau définie par l'IEEE en bas de la deuxième couche du modéle OSI. Elle gère notamment le partage du média entre plusieurs stations, et varie selon la technologie utilisée (Wi-Fi, Ethernet).

MAC — Message Authentication Code.

MySQL — MySQL est un système de gestion de base de données. Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle ou Microsoft SQL Server.

N

NAS — Network Access Server. Contrôleur d'accès au réseau dans l architecture RADIUS : lorsqu'un utilisateur cherche à se connecter au réseau via le NAS, celui-ci interroge le serveur RADIUS pour savoir s'il doit laisser passer l utilisateur, ou non.

NIS —Network Information Service (NIS) nommé aussi Yellow Pages est un protocole client serveur développé par Sun permettant la centralisation d'informations sur un réseau UNIX.

O

OID — pour Object Unique Identifier sont des identifiants universels, représentés sous la forme d'une suite d'entiers. Ils sont organisés sous forme hiérarchique. Ainsi seul l'organisme 1.2.3 peut dire quelle est la signification de l'OID 1.2.3.4. Ils ont été définis dans une

recommandation de l'International Telecommunication Union. L'IETF a proposé de représenter la suite d'entiers constituant les OID séparés par des points.

OpenSSH — (OpenBSD Secure Shell) est un ensemble d'outils informatiques libres permettant des communications sécurisées sur un réseau informatique en utilisant le protocole SSH.

OSI — Open Systems Interconnection. Conçu par l'ISO, le modèle OSI définit comment les protocoles réseaux doivent être organisés en couches superposées. Bien qu'il ne soit pas utilisé tel quel, le modèle OSI reste un modèle de référence.

P

PAP — Password Authentication Protocol est un protocole d'authentification pour PPP. Les données sont transmises en texte clair sur le réseau ce qui le rend par conséquent non sécurisé.

PGP — Le logiciel Pretty Good Privacy (ou PGP) est un logiciel de chiffrement et de signature de données utilisant la cryptographie asymétrique mais également la cryptographie symétrique. Il fait donc partie des logiciels de cryptographie hybride.

PKI — Public Key Infrastructure.

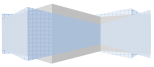
R

RFC — Les requests for comment, littéralement demande de commentaires, sont une série numérotée de documents électroniques documentant les aspects techniques d'Internet. Peu de RFC sont des standards, mais tous les standards d'Internet sont des RFC.

RADIUS — Remote Authentication Dial In User Service. Protocole de type AAA. Un réseau d'entreprise sécurisé par le WPA repose généralement sur un serveur RADIUS.

Roaming — Un accord de roaming entre deux opérateurs permet aux clients de l'un d'utiliser le réseau de l'autre.

RTS/CTS — Request to Send/Clear to Send. Lorsqu'un paquet de données doit être envoyé, si sa taille dépasse un seuil donné (le RTS Threshold), une requête RTS est d'abord envoyée pour demander la permission. Si le récepteur autorise l'envoi du paquet, il renvoie une réponse CTS à l'émetteur. Ce mécanisme permet de réduire les collisions dues aux stations qui ne sont pas à portée les unes des autres et ne peuvent donc pas savoir si elles risquent de prendre la parole en même temps.



S

SAML — Security assertion markup language (SAML) est un standard informatique définissant un protocole pour échanger des informations liées à la sécurité. Basé sur le langage XML, SAML a été développé par OASIS.

SASL — Simple Authentication and Security Layer (signifiant « Couche d'authentification et de sécurité simple » ou SASL) est un cadre d'authentification et d'autorisation normalisé par l'IETF.

SP — fournisseur de services (ou *Service Provider*, SP) : organisation membre d'une fédération et qui propose un service accessible via Internet : application web ou ressource numérique en ligne (pédagogique, scientifique, bibliothécaire, etc.). Le fournisseur de service n'a pas à gérer l'ensemble des utilisateurs susceptibles d'accéder au service. Il peut s'appuyer sur les fournisseurs d'identités de la fédération pour l'authentification des utilisateurs et pour obtenir les attributs des utilisateurs nécessaires au contrôle d'accès.

SSH — Secure Shell est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite toutes les trames sont chiffrées. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur.

SSID — Service Set Identifier (voir ESSID).

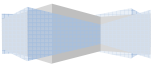
SSL — Secure Socket Layer.

SSO — L'authentification unique (ou identification unique ; en anglais Single Sign-On ou SSO) est une méthode permettant à un utilisateur de ne procéder qu'à une seule authentification pour accéder à plusieurs applications informatiques (ou sites internet sécurisés).

Station — Tout équipement capable de se connecter à un réseau (ordinateur, PDA).

T

TCP — Transmission Control Protocol (littéralement, « protocole de contrôle de transmissions ») est un protocole de transport fiable, en mode connecté, documenté dans la RFC 793 de l'IETF.



TLS — Transport Layer Security. Protocole permettant de mettre en place un tunnel sécurisé entre un client et un serveur. TLS est standardisé par l'IETF, et issu du protocole SSL conçu par Netscape.

TTLS — Tunneled TLS. Méthode d'authentification EAP très similaire à PEAP.

U

URN — Uniform Resource Name : identificateur d'une ressource. Un URN doit choisi de sorte d'être persistant, indépendant de la localisation de la ressource et pouvant être facilement mis en correspondance avec d'autres espaces de nommage. Le RFC 2141 définit la syntaxe des URN et cette page liste les RFCs relatifs aux URNs. Les entités d'une fédération (fournisseurs, attributs) sont désignés par des URNs.

UTF — UTF est un codage des caractères définis par Unicode où chaque caractère est codé sur un mot de bits.

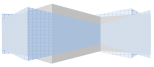
V

VLAN — Virtual LAN. Plusieurs réseaux virtuels peuvent être mis en œuvre sur une même infrastructure matérielle : chaque paquet contient alors un nombre (le VLAN ID) indiquant le VLAN auquel il appartient. Les VLAN sont mis en œuvre par des commutateurs et des AP compatibles avec la norme 802.1Q.

W

WAYF — service de découverte (ou *Discovery Service* ou *Where Are You From*, WAYF) : composant central dans une fédération qui permet à un utilisateur accédant à un service de sélectionner son organisation d'appartenance pour s'authentifier. Le service de découverte redirige l'utilisateur vers le service d'authentificaion de son organisation d'appartenance pour qu'il s'authentifie. Puis l'utilisateur authentifié est renvoyé vers le service voulu.

WECA — Wireless Ethernet Compatibility Alliance. Consortium fondé en 1999 en Californie entre plusieurs fabricants de matériel de réseau sans fil. Il s'est élargi et est devenu la Wi-Fi Alliance. Ce terme désigne toujours un comité est des normes utilisées pour certifier l'interopérabilité des produits Wi-Fi.

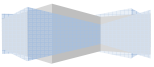


Wi-Fi — Certification de la Wi-Fi Alliance pour les produits respectant la norme 802.11.

Wi-Fi Alliance — Association de constructeurs de produits Wi-Fi.

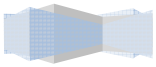
X

XML — Extensible Markup Language « langage extensible de balisage » est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents.

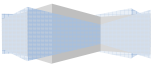


Références :

- [1] Guy Pujolle « 802.11 et les réseaux sans fil » publié en août 2002. Edition Eyrolles.
- [2] http://fédération.cru.fr/
- [3] http://www.chillispot.org/chilli.html, http://www.chillispot.org/FAQ.html
- [4] OASIS Security Services (SAML),
- www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
- [5] Serge Aumont, Claude Gross, Philippe Leca, Certificats X509 et infrastructure de gestion de clés. Dans Actes de la conférence JRES2001, Lyon, France, décembre 2001,
- [6] Stéphane Bortzmeyer, Les Web Services. Dans actes de la conférence JRES2003, Lille,France, décembre 2003, 2003.jres.org/TUTORIELS/paper.C.pdf.
- [7] The Shibboleth Project, shibboleth.internet2.edu.
- [8] RFC2616: Hypertext Transfer Protocol – HTTP/1.1,
- www.w3.org/Protocols/rfc2616/rfc2616.html.
- [9] Persistent client state, HTTP cookies, www.netscape.com/newsref/std/cookie_spec.html.
- www.cru.fr/igc/JRES01.tutoriel.IGC.pdf.
- [10] WiFi,déploiement et sécurité par Aurélien Géron, paru en 2006 AUX éditions Dunod
- [11] WAHL (M.), HOWES (T.) et KILLE (S.). –Lightweight Directory Access Protocol (v3) (LDAPv3) – IETF RFC 2251 (December 1997).
- [12] WAHL (M.), COULBECK (A.), HOWES (T.) et KILLE (S.). – Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions – IETF RFC 2252 (December 1997).
- [13] Recommandation UIT-T X.691 | ISO/ IEC 8825. – Spécification des Règles d'Encodage de Base (BER), des Règles d'Encodage Canoniques (CER) et des Règles d'Encodage Distinguées (DER) pour la Notation de Syntaxe Abstraite numéro Un (ASN.1) – (juillet 2002).
- [14] GRIMSTAD (A.), HUBER (R.), SATALURI (S.) et WAHL (M.). – Naming Plan for Internet Directory-Enabled Applications – IETF RFC 2377 (September 1998).



- [15] SMITH (M.). – Definition of the inetOrgPer- son LDAP Object Class – IETF RFC 2798 (April 2000).
- [16] BARKER (P.) et KILLE (S.). – The COSINE and Internet X.500 Schema – IETF RFC 1274 (November 1991).
- [17] WAHL (M.). – A Summary of the X.500 (96) User Schema for use with LDAPv3 – IETF RFC 2256 (December 1997).
- [18] WAHL (M.), HOWES (T.), et KILLE (S.). –Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distin- guished Names – IETF RFC 2253 (December 1997).
- [19] FERGEAU (F.). – UTF-8, a transformation format of ISO 10646 – STD 63, RFC 3629 (November 2003).
- [20] The Unicode Consortium – http:// www.unicode.org/standard/standard.html.
- [21] FREED (N.) et BORENSTEIN (N.). – Multi- purpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies – IETF RFC 2045 (November 1996).
- [22] Recommandation UIT-T X.680 | ISO/ IEC 8824. – Spécification de la Notation de Syntaxe Abstraite numéro Un (ASN.1) – (juillet 2002).
- [23] Information Sciences Institute, USC. –Transmission Control Protocol (TCP) – DARPA Internet Program, Protocol Specifi- cation RFC 793 (September 1981).
- [24] DIERKS (T.) et ALLEN (C.). – The TLS Proto- col Version 1.0 (TLS) – IETF RFC 2246 (January 1999).
- [25] ZEILENGA (K.) et CHOI (J.). – The LDAP Content Synchronisation Operation – IETF Internet Draft <draft-zeilenga-ldup-sync- 06.txt> (September 2004).
- [26] Comité Réseau des Universités – http:// www.cru.fr/ldap.



ملخص:

يتطرق هذا المشروع إلى انجاز ونشر شبكة لاسلكية أمنة داخل الحرم الجامعي, و ذلك على ثلاث مستويات:

- دراسة مخطط الجامعة، و نشر الهوائيات بهدف تحقيق التغطية الأمثل.

- دمج الحماية داخل الشبكة، وذلك بإعطاء كل المستخدمين شعار و هوية تمكنهم من دخول الشبكة.

إنجاز هذه الحماية تطلب استدعاء قاعدة بيانات و خادم توثيق و بوابة إلكترونية لحجز المعلومات.

- أخيرا، تركيب و تشكل لبنة للشعار و الهوية، و هي تكنولوجيا مستخدمة لإنشاء اتحادية تبادل وثائق إلكترونية بين الجامعات.

الكلمات المفتاحية:

الشبكة المحلية اللاسلكية ، واي فاي ، توفير هوية الشبكة المحلية اللاسلكية ، والأمن ، والشعار

Résumé :

Dans le cadre de notre projet de fin d'études, nous aborderons le déploiement d'un réseau sans fil sécurisé pour un campus universitaire sous un fournisseur d'identités Shibboleth.

La réalisation de ce projet passe par la mise en place de trois briques essentielles, à savoir :

- L'installation et la configuration des services dédiés à la communication sans fil.
- Le déploiement des antennes au niveau du campus
- En fin, l'introduction de la brique Shibboleth.

La première étape du projet, a été consacrée à la partie sans fil, pour ce faire, la solution envisagée s'appuie sur les standards (RADIUS, LDAP). Pour compléter cette première brique, un portail captif (chillispot (coovachilli)) a été mis en amont pour pouvoir identifier les différents utilisateurs.

Le serveur radius servira à l'authentification, par contre l'annuaire LDAP servira lui de base de données contenant toutes les données concernant les utilisateurs du réseau.

Par la suite, le déploiement des antennes se fera suivant un schéma permettant la couverture optimale du site universitaire afin d'assurer la mobilité et garantir une bonne qualité de service.

La dernière étape sera, l'installation et la configuration de la brique fournisseur d'identité Shibboleth, qui est une technologie utilisée pour instaurer un système fédéré de validation et d'autorisation sécurisée de l'identité.

Mots Clefs :
Réseau sans fil, Wifi, Fournisseur d'identité, Sécurité wifi, Shibboleth.

Abstract:

As part of our final project, we will discuss the deployment of a secure wireless network for a university campus in a Shibboleth Identity Provider.
This project involves setting up three bricks essential, namely:

- Installation and configuration of dedicated service to the wireless communication.
- The deployment of antennas at the campus.
- Finally, the introduction of brick Shibboleth.

The first phase was devoted to the security, to this end, the solution based on standards (RADIUS, LDAP). To complete this first brick, a captive portal (chillispot (coovachilli)) was set up to identify individual users.

The RADIUS server used for authentication by the LDAP against him serve database containing all data about users of the network.

Thereafter, the deployment of antennas will be following a pattern to the coverage of the university in order to ensure mobility and ensure a good quality of service.
The last step is the installation and configuration of the brick Shibboleth Identity Provider, which is a technology used to create a federated system of validation and authorization identity.

Keywords:
WLAN, WiFi, provider identity, WLAN security, Shibboleth.