République du Sénégal

Un peuple Un but Une foi

Université Cheikh Anta DIOP de DAKAR

Ecole Supérieure Polytechnique

Département Génie Informatique

Mémoire de fin de cycle pour l'obtention du
Diplôme **U**niversitaire de **T**echnologie

Thème :

Mise en place d'une solution de contrôle d'accès au réseau de l'UCAD

Lieu de stage : Direction Informatique de l'UCAD

Présenté et soutenu par : Mamadou SADIO et Savia Mt Mohamed ABDALLAH

Maître de stage : **Mme Khoudia SY**

Professeur encadreur : **Mme Khadidiatou Wane KEITA**

Année universitaire : 2008-2009

Sommaire

DEDICACES

Mamadou SADIO

Je dédie ce modeste travail :

- En la mémoire de mon père Douto SADIO qui nous a quittés il y a de cela six ans et qui, jusqu'à présent demeure constant dans mes pensées. A qui je ne dirai jamais assez merci pour l'éducation et le savoir qu'il nous a inculqués et qui constitue notre arme de survie. Paix à son âme et que la terre lui soit légère (amine),

- A ma raison d'être, celle sans qui, il n'y aurait pas vie, ma chère mère Sira DIATTA,

- A mes frères et sœurs qui ne cessent de me témoigner leur amour,

- Aux professeurs responsables de nos deux ans de formation ; en particulier à Mme KEITA, que j'appelais tout le temps « «maman » car en plus d'être notre encadreur lors de notre stage, c'est ce qu'elle représentait pour nous, notre promotion

- A mon maître de stage Mme Khoudia Guèye Sy ainsi qu'à toute son équipe.

- A tous mes camarades de promotion DUT 2007-2009 avec qui nous avons partagé joies et difficultés. Je les remercie pour le soutien moral qu'ils m'ont apporté tout au long de notre parcours.

- Ainsi qu'à tous mes amis.

Savia Mt MOHAMED ABDALLAHI

Je dédie ce travail à :

- Mon père Mohamed Abdallahi ould KAH (N'GOUDA) et ma mère Mariem vall Mt Ely ould MOCTAR : je ne pourrais jamais vous remercier assez pour tout ce que vous avez fait pour moi, pour toutes vos prières consenties à mon égard, votre dévouement pour ma réussite depuis ma naissance.

- Mes fréres Mohamed, Abdou, Abderahman et Mohameden.

- Mes sœurs Vatimetou et Esma pour leur sympathie et amour sans faille.

- Mon professeur encadreur Mme Khady Keita pour sa vision scientifique, la clarté dans ses choix et sa franchise.

- Mon maitre de stage Mme Khoudia Guéye Sy pour sa grande disponibilité.

- Mes Grandes mères, oncles et tantes Merci pour vos prières.

- A toutes mes amies en qui j'ai toujours trouvé le soutien et le réconfort.

- A Zahra et je lui dis merci pour les excellents moments qu'on a passé ensemble.

REMERCIEMENTS

Nous remercions tout d'abord ALLAH le tout puissant de nous avoir accordé vie et santé jusqu'à la réalisation de ce travail, ainsi que son prophète Mohamed (Paix et salut sur lui).

Nous tenons également à exprimer nos vifs remerciements à tous ceux qui par leur travail, leur idée, leur collaboration ont participé de prés ou de loin à la réalisation de ce mémoire.

Des remerciements particuliers :

- Au directeur de la direction informatique, Mr Samba NDIAYE et à son équipe, de nous avoir accueillis.

- Nous remercions également notre maîtresse de stage Mme Khoudia Guéye Sy, Chef de la Division Réseaux et Services Internet.
- Et à toute l'équipe de la Division Réseaux et Services Internet particulièrement à Mr Modou Diouf pour sa grande disponibilité et ses contributions, Mr Masamba Gaye et Mlle Amy Diouf.

- Au corps administratif et professoral du Département Génie Informatique de l'École Supérieure Polytechnique de Dakar.

- On ne saurait terminer sans exprimer tous nos remerciements à nos chers parents, pour le soutien qu'ils n'ont jamais cessé de nous apporter durant tout au long de nos études.

- On remercie Mr Gaye, chef du département informatique et tous les professeurs qui nous ont encadrés durant ce cycle.

- A toute la promotion DUT Informatique 2007-2009 de l'École Supérieure Polytechnique de Dakar pour leur esprit de groupe et leur parfaite entente.

Enfin à tous ceux qui ont participé de près ou de loin à la réalisation de ce projet, on vous dit MERCI.

PREAMBULE

L'École Supérieure Polytechnique (ESP) a été créée le 24 novembre 1994 suite aux recommandations de la concertation nationale sur l'enseignement supérieur qui s'est tenue d'avril 1992 à août 1993 et qui préconisait la reconstruction des écoles d'enseignement technologique de l'UCAD.

Elle regroupe la division industrielle de l'ex-ENSUT (École Nationale Supérieure Universitaire de Technologie) ; la section des sciences et techniques industrielles de l'ex ENSEPT (École Nationale Supérieure d'Enseignement Professionnelle et Technique) et l'ex EPT (École Polytechnique de Thiès).
L'ESP est une institution regroupant deux centres :

- Le centre de Dakar (ex-ENSUT /ENSEPT)
- Le centre de Thiès (ex-EPT)

L'École comporte aujourd'hui six (6) départements

- Le département GENIE INFORMATIQUE (centre de Dakar)
- Le département GENIE CIVIL (centre de Thiès)
- Le département GENIE ELECTRIQUE (centre de Dakar)
- Le département GENIE MECANIQUE (centre de Dakar)

- Le département GENIE CHIMIQUE (centre de Dakar)
- Le département GESTION (centre de Dakar)

L'ESP forme tant sur le plan théorique que pratique des :

- Techniciens supérieurs (DUT et DST)
- Ingénieurs de conception (DIC)
- Ingénieurs d'exécution sur la demande des entreprises (DIT)

Le Département Génie Informatique propose un enseignement :

- fondamental, pour acquérir des connaissances, des concepts de base et des méthodes de travail,
- appliqué pour faciliter l'apprentissage de ces concepts et déployer des savoir faire professionnel,
- évolutif pour intégrer les projets technologiques et les exigences du monde professionnel,
- ouvert, pour développer les facultés de communication indispensables aux informaticiens dans l'exercice de leur métier.

Dans le cadre de leur formation, chaque étudiant du cycle DUT du département informatique, est tenu d'effectuer un stage pratique dans un service ou une société de la place.

Ce stage devra permettre à l'étudiant :

- de concevoir un système et de mener à bien l'élaboration de celui-ci depuis l'étude préalable jusqu'à sa mise en exploitation.
- De mettre en œuvre les connaissances théoriques et pratiques acquises tout au long de leur formation.
- d'être confrontés aux réalités du milieu professionnel et de se faire la main sur des projets d'envergure.

Au terme du stage, l'étudiant doit rédiger un mémoire sur le problème informatique qu'il a étudié et auquel il a essayé de trouver des solutions appropriées.

C'est dans ce cadre que nous avons été amenés à effectuer un stage pratique d'une durée moyenne de deux (2) mois au sein de la Direction Informatique de l'UCAD.

INTRODUCTION

Le réseau informatique est un ensemble d'équipements reliés entre eux pour échanger des informations. Ils ont fait irruption dans le quotidien des entreprises et structures permettant ainsi une amélioration des services qui y sont offerts entre autres le partage de ressources, la gestion centralisée de ces ressources, meilleure circulation des informations au sein de l'entreprise ou la structure…

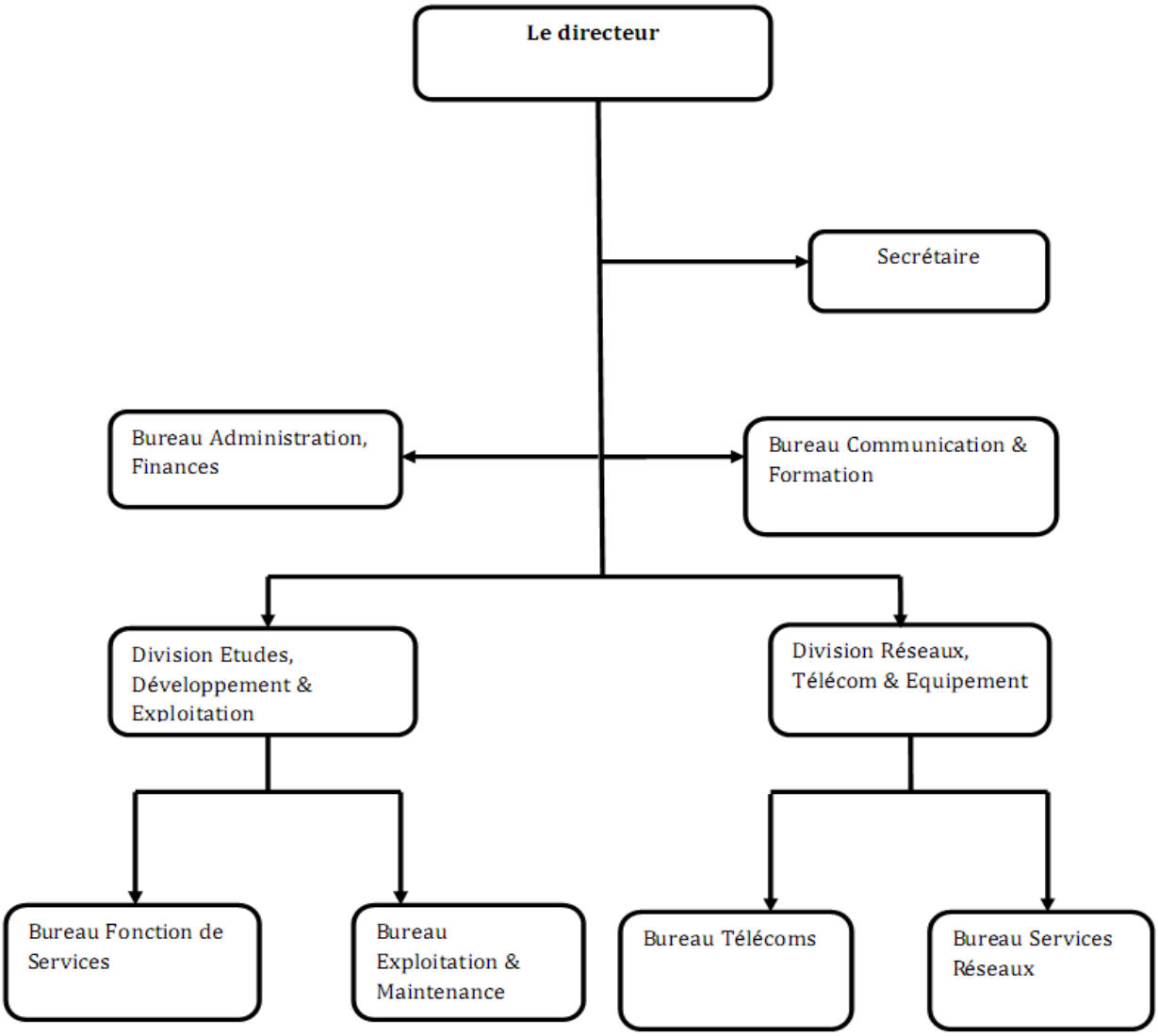
La sécurisation et le contrôle de l'accès à ces réseaux s'imposent car il y a certainement, au sein de ce réseau, des informations auxquelles ne doivent pas avoir accès les tiers tout comme n'importe qui n'a pas le droit d'accéder aux ressources du réseau. C'est dans ce sens que s'oriente notre sujet de stage qui s'énonce comme suit : Mise en place d'une solution de contrôle des accès au réseau informatique de l'Université Cheikh Anta DIOP.

Nous allons d'abord présenter notre cadre de travail, à travers laquelle nous allons expliquer le motif du sujet dans la problématique, puis faire l'inventaire des solutions déployables dans la seconde partie, ensuite faire une étude comparative pour aboutir au choix de celle, jugée optimale qui résoudrait notre problème enfin passer à sa mise en œuvre.

I. PRESENTATION DU CADRE DE TRAVAIL

1. ORGANIGRAMME

La direction informatique de l'université Cheikh Anta Diop de Dakar (UCAD), créée en 1999 a comme principale activité la gestion du système d'information de l'université Cheikh Anta DIOP de Dakar.


Le directeur
Secrétaire
Bureau Administration, Finances
Bureau Communication & Formation
Division Etudes, Développement & Exploitation
Division Réseaux, Télécom & Equipement
Bureau Fonction de Services
Bureau Exploitation & Maintenance
Bureau Télécoms
Bureau Services Réseaux

2. MISSIONS DE LA DIRECTION INFORMATIQUE DE L'UCAD

La direction informatique est chargée :

- ✓ d'organiser l'architecture de l'internet et de la messagerie à l'UCAD ;
- ✓ de gérer le réseau et son extension, d'aider les structures à connecter tous les bureaux ;
- ✓ de gérer les outils pour l'utilisation optimale de cet environnement ;
- ✓ de tenir à jour le site Web de l'UCAD en indiquant les activités hebdomadaires ;
- ✓ d'élaborer un programme de formation des enseignants en informatique ;
- ✓ d'étudier et de mettre en œuvre un projet de formation en informatique des PATS, en rapport à la DRH ;
- ✓ de gérer la télécommunication interne de l'UCAD.

Afin d'atteindre ses objectifs, la Direction s'est structurée en deux divisions et six bureaux et pour chacune de ses composantes, un agent a été désigné responsable, mais dans la réalité le travail à tous les niveaux se fait en équipe :

Les deux premiers bureaux sont directement rattachés au directeur de la DI :

Bureau Formation et Communication

- ✓ Organiser dans l'espace Universitaire, une communication efficace entre la direction de l'informatique et ses composantes internes de la communauté ;
- ✓ Aider à mener les études pour la réalisation d'un réseau téléphonique interne ;
- ✓ Concevoir, réaliser ou faire réaliser des programmes de formation aux NTIC pour le personnel de L'UCAD ;
- ✓ Elaborer des programmes de formation des enseignants en Informatique ;
- ✓ Organiser des sessions de formation spécialisées pour les personnels de l'UCAD ;
- ✓ Aider à trouver des opportunités pour la fonction de service.

Administration et Finances :

- ✓ Procéder à l'engagement, à la liquidation, à la certification des dépenses à effectuer ;
- ✓ Effectuer les taches administratives et financières, conformément à la réglementation en vigueur ;
- ✓ Entrer en contact avec les fournisseurs pour les contacts de maintenance et d'entretien des équipements informatiques ;
- ✓ Passer les écritures comptables et arrêter les droits des créanciers ;
- ✓ Gérer l'intendance de la direction.

❖ Division Etudes et Développement

Elle se charge :

- ✓ d'assurer la conception et le développement de nouveaux projets (Paie, Gestion des examens) ;
- ✓ assurer la veille technologique ;
- ✓ maintenir la cohérence du système d'information de l'UCAD.

Elle prend en charge deux bureaux que sont :

Bureau Fonction de service :

- ✓ Concevoir et mettre en œuvre des fonctions de service dans les domaines de la formation du développement d'application et de la consultation, dans le but d'accroitre les revenus de la direction ;
- ✓ Etre responsable de la salle de formation et des équipements annexes ;
- ✓ Etre responsable de la salle de développement de logiciels.

Le Bureau Fonction de service est rattaché à la division Etudes et développement.

Bureau Exploitation :

- ✓ Assurer le fonctionnement quotidien des applications de gestion, notamment les conditions de haute disponibilité ;

- ✓ Assurer la sauvegarde des données ;
- ✓ Assurer le support utilisateur.

❖ **Division Réseau, Télécoms et Equipements :**

- ✓ administrer le réseau informatique et en assurer la haute disponibilité ;
- ✓ gérer la sécurité du réseau ;
- ✓ gérer l'accès Internet, y compris l'accès distant ;
- ✓ gérer la messagerie Internet ;
- ✓ gérer le domaine « **ucad.sn** » ;
- ✓ aider à la réalisation d'un réseau téléphonique interne ;

Elle aussi prend en charge deux bureaux :

Bureau Télécom :

- ✓ Gérer les équipements ;
- ✓ Gérer la connexion Internet ;
- ✓ Gérer la connexion des structures.

Bureau Services réseau :

- ✓ Assurer le fonctionnement quotidien des services tels que :
- ✓ Web
- ✓ DNS
- ✓ MAIL
- ✓ TELEPHONE
- ✓ VISIOCONFERENCE
- ✓ ……

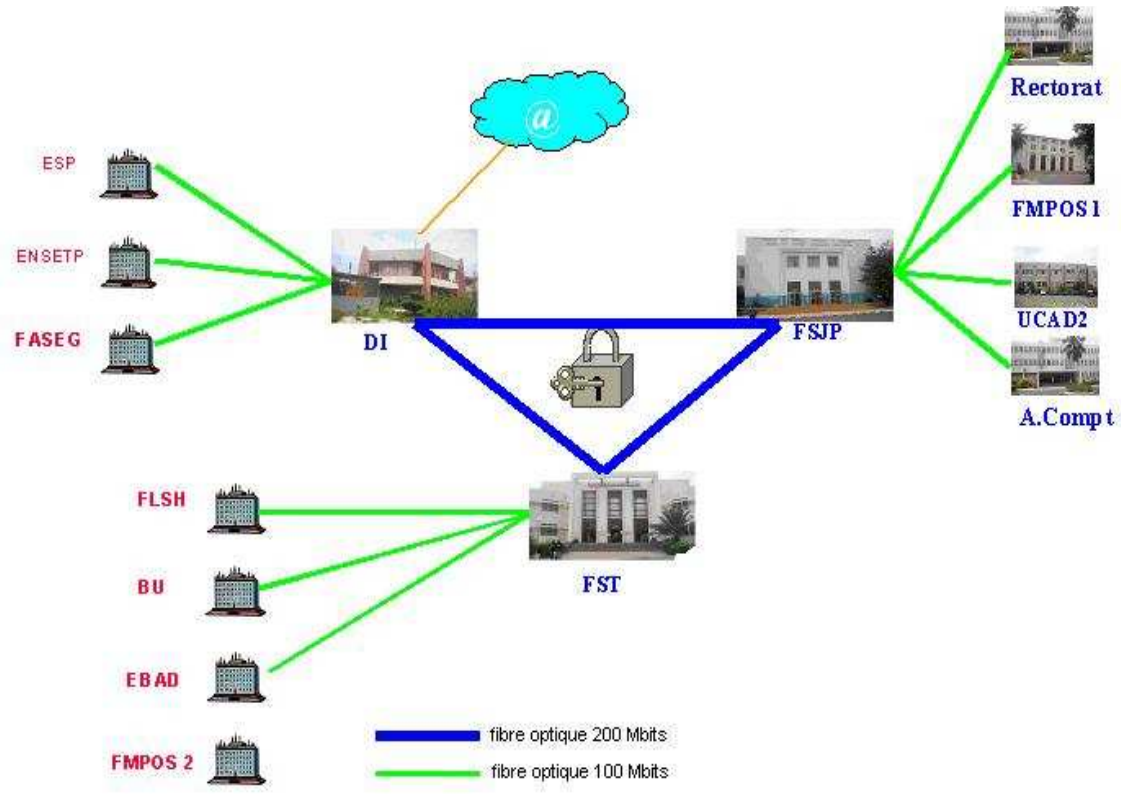
3. PRESENTATION DU RESEAU DE L'UCAD

Le réseau informatique de l'Université Cheikh Anta Diop (UCAD) est un grand réseau de type MAN (Metropolitan Area Network) qui s'appuie sur un backbone en triangle et

couvre l'ensemble des établissements (cinq facultés, cinq écoles et instituts) et les structures entre autres le rectorat, agence comptable qui lui sont rattachés. Il s'étend sur une longueur d'environ 3Km et les composants du réseau sont reliés par des fibres optiques ou par ondes hertzienne. **95%** des équipements du réseau comme les routeurs et les commutateurs sont de marque Cisco. L'UCAD possède deux liaisons spécialisées de deux (2) Mégabits et trois (3) ADSL de dix Mégabits Max ce qui fait en tout une connexion de 34 Mégabits pour assurer la liaison Internet. Mais le point de connexion vers l'Internet ne fait l'objet de notre rapport.

En effet notre travail sera essentiellement accentué sur le réseau filaire Intranet de l'UCAD comme l'illustre ce schéma qui suivant :

Le réseau filaire :



4. PROBLEMATIQUE

Comme il a été tantôt dit, la gestion du réseau de l'UCAD est centralisée à la D I. De ce fait une surveillance étroite ne peut pas être faite sur l'accès au réseau. Bien vrai que des efforts ont été fournis par la division réseau, allant dans le sens de l'optimisation du réseau en le segmentant en sous-réseaux virtuels (Virtual Local Area Network VLAN), et une restriction d'accès à certains sites web, cela ne leur permet pas de contrôler l'accès physique au leur réseau.

Aucune politique de contrôle d'accès n'a été mise en place, raison pour laquelle n'importe qui possédant une machine peut accéder au réseau en se connectant sur une prise mural, et avoir une connexion à l'Internet. Et cela ne manque pas d'impacts sur la bande passante et surtout sur la sécurité du réseau.

En effet, l'université possède des applications entres autres, pour l'administration qui sont au nombre de trois, développées par la direction informatique qui permettent de faire:

- la gestion financière (budget et comptabilité)
- la gestion de la scolarité (inscription administrative et pédagogique des étudiants, emploi du temps des salles et classes, gestion des examens)
- la gestion des ressources humaines (recrutements, congés, avancements).

Elles sont critiques car manipulant des données auxquelles ne doivent pas avoir accès les tiers, d'où le souci de confidentialité.

Cependant tenant compte de la taille du réseau, il serait fastidieux, aussi bien pour nous que pour l'administrateur du réseau, de procéder par une identification personnelle de tous les utilisateurs du réseau en leur procurant à chacun login et mot de passe.

Face à ce problème, ce qu'il faudrait, c'est de mettre en place un système de contrôle des accès qui réglementerait les connexions au réseau.

II. ETAT DE L'ART SUR LES SOLUTIONS DE CONTROLE D'ACCES AU RESEAU

Le contrôle d'accès consiste à vérifier si une entité (une personne, un ordinateur, …) demandant d'accéder à une ressource a les droits nécessaires pour le faire.

Un contrôle d'accès offre ainsi la possibilité d'accéder à des ressources physiques (par exemple un bâtiment, un local, un pays) ou logiques (par exemple un système d'exploitation ou une application informatique spécifique).

Le contrôle d'accès vise à intercepter toutes les tentatives d'accès aux ressources ou aux informations critiques. Les politiques de sécurité définissent les règles de haut niveau qui régissent les accès. Les mécanismes de sécurité définissent les fonctions de bas niveau (logicielles et matérielles) permettant d'implanter le contrôles imposés par la politique. Le contrôle peut être réalisé à l'aide de l'utilisation d'éléments permettant l'authentification ou l'identification de l'entité (par exemple un mot de passe, une carte, une clé, un élément biométrique, …).

Cette partie fera l'objet d'une étude globale des outils pouvant prendre en charge le contrôle des accès au réseau. Les serveurs connus sont VMPS et RADIUS.

Cependant ils n'ont pas les mêmes manières de procéder, toutefois chacun utilise des protocoles et un gestionnaire de données.

A. LE SERVEUR VMPS

1. PRE-REQUIS

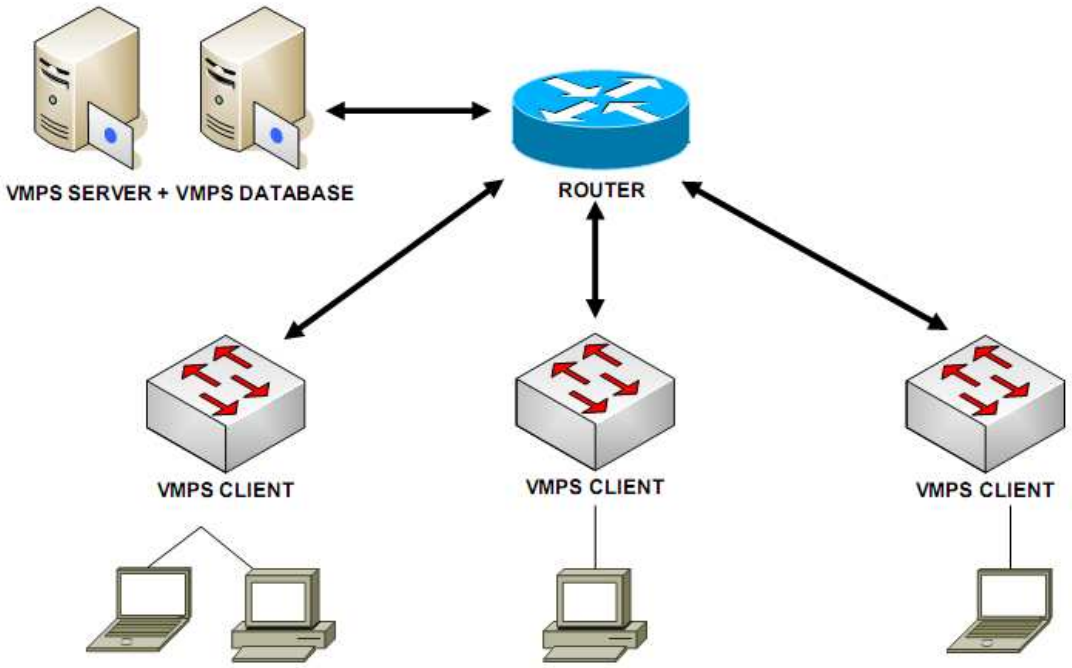
VMPS (Vlan Management Policy Server) est un service propriétaire Cisco qui permet d'associer dynamiquement une ou plusieurs adresses MAC (Media Access Control ou adresse physique de l'hôte) à un Vlan. Ce faisant, les commutateurs clients doivent être de marque Cisco et doivent être compatibles "client VMPS". La gestion dynamique des Vlans passe tout d'abord par une bonne maîtrise d'une architecture réseau basé sur les Vlans de niveau 2 (association Vlan/adresse MAC), du protocole VTP (Vlan Trunking Protocol), puis par l'étude du protocole VQP (Vlan Query Protocol). Nous avons deux options de configuration du serveur :

- prendre un commutateur (Cisco bien sûr) comme serveur ;
- utiliser la version « open source » (dont le code source est consultable et modifiable) destiné aux plateformes Unix (Linux et Solaris) et le plus connu et le plus utilisé est OpenVMPS.

Il est conseillé d'avoir un serveur secondaire qui va assurer la redondance en cas de panne.

Il faut nécessairement avoir un serveur TFTP ou l'équivalent où seront stockées les informations relatives aux hôtes qui se connectent dans le réseau.

2. PRINCIPE DE FONCTIONNNEMENT



Le serveur VMPS télécharge une base de données contenant les associations adresse MAC et le nom du vlan correspondant depuis un serveur TFTP (Trival File Transfer Protcol), c'est à partir de ce moment que le serveur devient prêt à répondre aux requêtes des hôtes qui se connectent. Cette base de données est téléchargée à chaque fois que le serveur est démarré ou redémarré.

Le VMPS ouvre un socket, c'est à dire un espace mémoire réservé par l'application ouvert sur le réseau à travers lequel il accueille les connexions entrantes et sortantes. Dans ce cas, c'est le protocole UDP qui est utilisé, à l'opposé de TCP. Certes cela ne garantit pas l'acheminement des paquets mais en revanche est beaucoup plus rapide.

Lorsque le client envoie une requête valide au serveur, ce dernier recherche dans sa base de données, l'adresse MAC (Media Access Control) correspondante et le numéro de Vlan auquel elle est associée. Si le Vlan choisi est autorisé sur le port, le nom du Vlan est envoyé au client (le Switch). Si le Vlan n'est pas autorité sur le port, deux cas peuvent se produire :

- si le serveur VMPS est en mode « open », un message avertissant que l'accès au réseau est refusé est renvoyé au client,

- ou si le serveur VMPS est en mode « Secure », le port du Switch est alors refermé.

S'il y a des adresses MAC dans le réseau auxquelles nous ne souhaitons pas donner l'accès aux ressources du réseau, une option permet, dans la configuration du serveur, de les filtrer en ne leur spécifiant aucun Vlan. C'est l'option -NONE- qui doit se trouver à la place réservée usuellement au nom du Vlan. Dans ce cas, le serveur réagit selon le mode de configuration que nous avons fait (mode « open » ou mode « Secure »).

Nous avons aussi la possibilité de créer des groupes de ports associés à un Vlan. Lors d'une connexion sur un de ces ports, le Vlan est récupéré dans la base de données en fonction de l'adresse MAC puis le serveur le compare à celui associé aux ports:

- Si les deux noms de Vlan concordent, le port est ouvert et placé dans le bon Vlan.
- Sinon, le serveur envoie un message annonçant que l'accès est refusé ou referme le port si le serveur est en mode « Secure ».

Un port en configuration dynamique ne peut donc être associé qu'à un seul Vlan. Les connexions multiples sur un port (via un doubleur ou un hub) ne sont possible que pour les adresses MAC des machines se trouvant dans le même Vlan. Si un port est déjà affecté à un Vlan et qu'une nouvelle demande de connexion, dans un autre Vlan, a lieu, le serveur envoie un message annonçant que l'accès est refusé ou referme le port si le serveur est en mode " Secure ".

Voici, de manière très résumée, les étapes d'une connexion :

1 - L'utilisateur nouvellement connecté (ou qui vient de changer de port sur le commutateur) envoi un paquet au commutateur client VMPS.

2 - Avec ce paquet, le commutateur apprend l'adresse MAC associée à la station et l'associe à un port.

3 - Le commutateur client VMPS envoie une requête au serveur VMPS qui contient l'adresse IP du commutateur client, l'adresse MAC de la station, le port où la station est raccordée, le domaine VTP s'il y en a un.

4 - Le serveur VMPS lit le fichier de configuration et vérifie si l'adresse MAC figure bel et bien dans la base de données.

5 - Le serveur VMPS envoie une requête en guise de réponse au commutateur client VMPS.

6 - La décision d'affectation d'un numéro de VLAN est faite selon la réponse envoyée à l'étape 5 (donner l'accès ou le lui refuser).

Avec le serveur VMPS, il ne nécessite pas une configuration particulière au niveau des stations de travail.

Remarque : Les routeurs fixes, modulaires ou ISR n'ont pas de fonctionnalités VMPS que ce soit client ou serveur. VMPS fonctionne uniquement sur des commutateurs et ceux pouvant faire office de serveur VMPS sont les séries:

Catalyst 5000, 5500,

Catalyst 6000, 6500.

3. LES PROTOCOLES

a. LE PROTOCOLE VTP (Vlan Trunking Protocol)

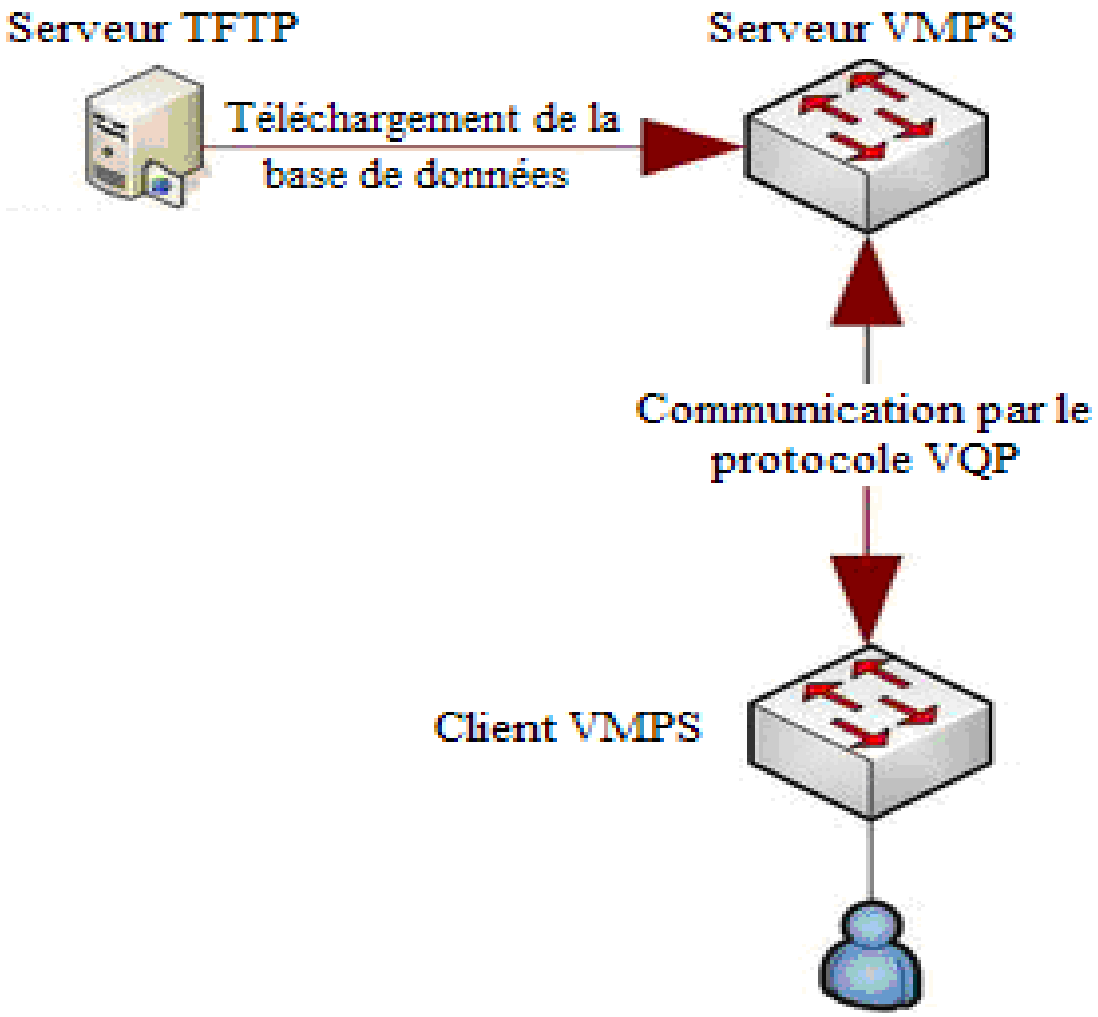
Un **trunk** est une liaison physique unique à travers laquelle est véhiculé tout le trafic des réseaux locaux virtuels (VLAN). Les trames traversant le trunk sont complétées avec un identificateur de réseau local virtuel (VLAN id) : on dit que la trame est « taggée » par le commutateur qui l'a envoyée. Et c'est cette identificateur qui permet la conservation des trames au sein d'un même VLAN ou leur domaine de diffusion.

L'utilisation du protocole VTP est indispensable sur le réseau sur lequel nous voulons déployer une solution VMPS. En effet, ce dernier se servira du nom de domaine VTP lors de l'échange de paquets entre le serveur VMPS et les commutateurs clients.

b. LE PROTOCOLE VQP (Vlan Query Protocol)

Le protocole VQP est un protocole qui permet au Switch client d'interroger un serveur VMPS avec des informations sur les stations enregistrées et leur Vlan associé. C'est ce qui permet au Switch client de pouvoir associer le port au Vlan correspondant.

Le VMPS est donc basé sur une architecture client/serveur et permet de gérer dynamiquement les assignations de Vlan en fonction d'adresses MAC. Lorsqu'une machine se connecte à un port, le Switch récupère son adresse MAC et se connecte au serveur VMPS afin de vérifier le droit d'accès de cette machine. Lorsque celle-ci est autorisée, le serveur envoie au client VMPS le Vlan dans lequel cette machine doit se connecter. Le Switch place donc le port dans le bon Vlan et la machine a donc accès au réseau.



Il est préférable que le serveur VMPS soit accompagné d'un serveur secondaire pour prendre le relais en cas de panne ou pour simplement équilibrer la charge.

Le serveur utilise une base de données contenant les adresses MAC des clients, le Vlan correspondant et d'autres règles permettant de donner le droit à un utilisateur de se connecter.

4. LES GESTIONNAIRES DE DONNEES

Pour les gestionnaires de données, nous avons la possibilité d'utiliser soit un serveur TFTP, les bases de données classiques comme MySQL, ou un annuaire électronique.

a. LE SERVEUR TFTP

TFTP (pour ***Trivial File Transfer Protocol*** ou **Protocole simplifié de transfert de fichiers**) est un protocole simplifié de transfert de fichiers.

Il fonctionne sur le port 69 UDP, au contraire du FTP (File Transfer Protocol) qui utilise lui TCP (Transmission Control Protocol). L'utilisation d'UDP, protocole « non fiable », implique que le client et le serveur doivent gérer eux-mêmes une éventuelle perte de paquets. En termes de rapidité, l'absence de fenêtrage nuit à l'efficacité du protocole sur les liens à forte latence. On réserve généralement l'usage du TFTP à un réseau local.

Les principales simplifications visibles du TFTP par rapport au FTP est qu'il ne gère pas le listage de fichiers, et ne dispose pas de mécanismes d'authentification, ni de chiffrement. Il faut connaître à l'avance le nom du fichier que l'on veut récupérer. De même, aucune notion de droits de lecture/écriture n'est disponible en standard.

À cause de ces fonctionnalités absentes, FTP lui est généralement préféré. TFTP reste très utilisé pour la mise à jour des logiciels embarqués sur les équipements réseaux (routeurs, pare-feu, etc.).

A l'équivalence d'un serveur TFTP, VMPS peut utiliser un gestionnaire de données, où il stocke les informations relatives aux entités qui demandent à accéder aux ressources du réseau, qui peut être une base de données classique (MySQL) ou un annuaire électronique (LDAP).

b. LDAP

Un annuaire électronique est aussi une base de donnée, mais spéciale, ayant comme fonction première de retourner un ou plusieurs attributs d'un article à l'aide de fonctions de recherche multicritères. Il permet le partage des bases d'informations sur le réseau, qu'il soit interne ou externe. Les informations contenues dans ces bases peuvent être très variées ; Exemple : les coordonnées de personnes ou des données systèmes ou en ce qui nous concerne, des données relatives à un hôte. Un annuaire électronique peut servir d'entrepôt permettant une centralisation des informations les rendant du coup disponibles, via le réseau, à des applications, des systèmes d'exploitation ou des utilisateurs. L'annuaire électronique le plus utilisé dans les services réseau est LDAP dont nous allons faire la présentation.

b.1 PRESENTATION

LDAP (Lightweight Directory Access Protocol) est un service d'annuaire conçu à partir de la norme X.500 qui aussi en était une. La norme X.500 est très lourde, LDAP en est une version allégée ("light") dans un sens absolument pas péjoratif. Les serveurs LDAP possèdent une grande capacité de stockage avec des données de faible volume et permettent un accès en lecture très rapide à celles-ci grâce au modèle hiérarchique.

Voici les concepts de LDAP :

- le protocole permettant d'accéder à l'information contenue dans l'annuaire,
- un modèle d'information définissant le type de données contenues dans l'annuaire,
- un modèle de nommage définissant comment l'information est organisée et référencée,
- un modèle fonctionnel qui définit comment on accède à l'information ,
- un modèle de sécurité qui définit comment données et accès sont protégés, OpenLDAP est souvent configuré avec SASL (Simple Authentication and Security Layer), qui permet les transactions cryptées.
- avec les protocoles fonctionnant en mode connecté
- un modèle de duplication qui définit comment la base est répartie entre serveurs,
- des APIs pour développer des applications clientes
- LDIF (LDAP Data Interchange Format) qui est le format utilisé pour l'échange de données.

LDAP utilise le jeu de caractères *Unicode Transformation Format- 8* (UTF-8) pour le stockage des valeurs d'attributs de type texte et celui des DNs. UTF- 8 englobant tous les jeux de caractères (isoLatin, Shift- JLS...), on peut employer différentes langues pour les valeurs d'attribut grâce à l'option *language code* de l'attribut (extension proposée par l'IETF). On peut donc ainsi créer des annuaires multilingues.

Par exemple, on peut avoir pour un objet personne, un attribut *description* en français et un autre en japonais :

description, lang-fr : ***le texte en français***
description, lang-ja : ***le même en japonais***

Ce qui va nous permettre de personnaliser notre annuaire suivant la langue que nous désirons. Les règles qui régissent le fonctionnement de LDAP sont décrites dans son protocole.

b.2 LDAP, LE PROTOCOLE

C'est un protocole qui définit la manière dont s'établit la communication entre le client et le serveur. Il fournit à l'utilisateur des commandes lui permettant de se connecter ou se déconnecter , pour rechercher, comparer, créer, modifier ou effacer des entrées. C'est un protocole fiable et sécurisé car utilisant des mécanismes de chiffrement connus comme :

- SSL (Secure Sockets Layer) ou TLS (Transport Layer Security) pour le transfert et l'accès aux données,
- SASL couplés avec des mécanismes de règles d'accès comme ACL (Access Control List), pour d'authentification.

Voici quelques commandes utilisées pour effectuer des opérations de base :

Opération	Description
Search	Recherche dans l'annuaire d'objets
Compare	Comparaison du contenu de deux objets
Add	Ajout d'une entrée
Modify	Modification du contenu d'une entrée
Delete	Suppression d'un objet
Rename	(Modify DN) modification du DN d'une entrée
Bind	Connexion au serveur
Unbind	Deconnexion

Les opérations de type « search » ou « compare » reçoivent des paramètres :

Paramètres	Description
base object	l'endroit de l'arbre où doit commencer la recherche
scope	la profondeur de la recherche

size limit	nombre de réponses limite
time limit	temps maximum alloué pour la recherche
attrOnly	renvoie ou pas la valeur des attributs en plus de leur type
search filter	le filtre de recherche
list of attributes	la liste des attributs que l'on souhaite connaître

La gamme de logiciels serveurs LDAP propose également un protocole de communication entre serveurs, leur permettant d'échanger leur contenu et de le synchroniser (replication service) ou de créer entre eux des liens permettant ainsi de relier des annuaires les uns aux autres (refferal service) .

b.3 FORMAT DE LA BASE

b.3.1 LE DIT (DIRECTORY INFORMATION TREE)

C'est sous forme hiérarchique que LDAP organise les données qu'il prend en charge. L'arbre est nommé « Directory Information Tree » (**DIT**). Le sommet (racine), contient le "suffixe". Chaque nœud représente une « entrée » ou « Directory Entry Service » (**DSE**). L'entrée située à la racine est appelée « rootDSE » (root Directory Entry Service), qui décrit la structure de l'arborescence (le **DIT**) ainsi que son contenu. Chaque entrée est connue de manière unique dans l'arborescence grâce à son **dn** (Distinguished Name). Le dn indique le chemin à parcourir pour en partant du sommet arriver à l'entrée correspondante. Par exemple pour identifier une machine, on part du pays (sn), puis le nom de domaine (ucad pour la suite des opérations), le VLAN auquel il appartient et enfin le nom de la machine ou son adresse physique, l'ensemble de ces paramètres est le **dn** qui identifie de manière unique une machine.

Les données sont stockées sur un format de base de données hiérarchique de type « dbm ». Ce format est différent des bases de données relationnelles, conçues pour supporter de multiples mises à jour. DBM est conçu pour supporter peu de mises à jour, mais de nombreuses consultations.

B.3.2 LES ATTRIBUTS

Chaque entrée peut être considérée comme un objet (au sens de l'orienté objet) possédant donc certains attributs, par exemple si nous prenons une machine comme entrée, les attributs peuvent être, le nom, son adresse physique, son adresse IP, ... On peut aussi définir des attributs obligatoires et d'autres optionnels, en d'autres termes, les attributs obligatoires devront être renseignés mais pas forcément les optionnels. Il existe par ailleurs pour chaque **DSE** des attributs d'administration qui ne servent qu'au serveur.

B.3.3 LES CLASSES D'OBJETS

On regroupe les objets qui sont du même domaine dans une classe d'objects. Une classe d'objet est définie par un nom, un OID (Object IDentifier), la liste des attributs (facultatifs et obligatoires), un type. Les types de classe d'objet sont:

- type structurel car elle contient des d'objets concrets de l'annuaire (personnes, groupes de personnes, ...) ;
- type auxiliaire, ce sont des classes d'objets que nous pouvons créer, pour rajouter des informations (attributs) supplémentaires à des classes d'objets de type structurel déjà existantes. En orienté objet, on dira que la classe auxiliaire hérite d'une classe structurelle,
- type abstraite, ce sont les classes d'objets qui existent par défaut et qui n'ont pas de signification concrète, par exemple la classe **top** est la classe d'objet générique. Toutes les autres classes héritent de cette classe.

Le principe est donc le même qu'en orienté objet, on retrouve une structure arborescente, avec à la racine la classe **top**, toutes les autres classes d'objets dérivent de cette classe générique, chaque classe hérite des propriétés d'une classe mère et possède des attributs supplémentaires par rapport à cette dernière.

B.3.4 LES SCHEMAS

Un schéma représente l'ensemble de la description des classes d'objets et des types d'attribut.

B.3.5 LE FORMAT D'ECHANGE DE DONNEES LDIF

LDIF (LDAP Data Interchange Format) permet de représenter les données LDAP sous format texte standardisé, il est utilisé pour afficher ou modifier les données de la base. Il est destiné à donner une lisibilité des données par les utilisateurs.

LDIF offre deux optiques :

- Il permet d'importer ou d'exporter une base de données
- ou de faire l'ajout et des modifications sur des entrées.

Les données sont en ASCII codées en UTF−8, sauf pour le binaire qui est codé en base64 (images par exemple).

c. MySQL

MySQL, le plus populaire des serveurs de bases de données SQL Open Source, a été développé, distribué et supporté par MySQL AB. MySQL AB est une société commerciale, fondée par les développeurs de MySQL, qui développent leur activité en fournissant des services autour de MySQL. Cette société a été achetée par Sun Microsystems le 16 janvier 2008.

- MySQL est un système de gestion de bases de données.

Une base de données est un ensemble organisé de données. Cela peut aller d'une simple liste de courses au supermarché à une galerie de photos, ou encore les grands systèmes d'informations des multinationales. Pour ajouter, lire et traiter des données dans une

base de données, vous avez besoin d'un système de gestion de bases de données tel que le serveur MySQL. Comme les ordinateurs sont très bons à manipuler de grandes quantités de données, le système de gestion de bases de données joue un rôle central en informatique, aussi bien en tant qu'application à part entière, qu'intégré dans d'autres logiciels.

- MySQL est un serveur de bases de données relationnelles.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans ``MySQL'' signifie ``Structured Query Language'' : le langage standard pour les traitements de bases de données.

- MySQL est Open Source.

Open Source (Standard Ouvert) signifie qu'il est possible à chacun d'utiliser et de modifier le logiciel. Tout le monde peut télécharger MySQL sur Internet, et l'utiliser sans payer aucun droit. Toute personne en ayant la volonté peut étudier et modifier le code source pour l'adapter à ses besoins propres. Le logiciel MySQL utilise la licence GPL (GNU General Public License). Il fonctionne aussi sur plusieurs systèmes d'exploitation incluant AIX, IBM i-5, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000, XP et Vista.

- Le serveur de bases de données MySQL est très rapide, fiable et facile à utiliser

Le serveur de bases de données MySQL dispose aussi de fonctionnalités pratiques, développées en coopération avec les utilisateurs.

Le serveur MySQL a été développé à l'origine pour gérer de grandes bases de données plus rapidement que les solutions existantes, et a été utilisé avec succès dans des environnements de production très contraints et très exigeants, depuis plusieurs années. Bien que toujours en développement, le serveur MySQL offre des fonctions nombreuses et puissantes. Ses possibilités de connexions, sa rapidité et sa sécurité font du serveur MySQL un serveur hautement adapté à Internet.

B. LE SERVEUR RADIUS

Contrairement au serveur VMPS, RADIUS (Remote Authentication Dial-In User Service) n'a pas d'exigences particulières. Néanmoins son principe de fonctionnement diffère un peu de celui de VMPS.

1. PRINCIPE DE FONCTIONNENMENT

RADIUS est un protocole client/serveur destiné à permettre à des serveurs d'accès de communiquer avec une base de données (ou son équivalent) centralisée regroupant en un point l'ensemble des utilisateurs distants ou locaux. Il repose principalement sur un serveur (le serveur RADIUS), relié à une base d'identification (base de données, annuaire LDAP, etc.), sur un client RADIUS, appelé NAS (Network Access Server), faisant office d'intermédiaire entre l'utilisateur final et le serveur. L'ensemble des transactions entre le client RADIUS et le serveur RADIUS est chiffré. Le serveur RADIUS peut faire office de proxy, c'est-à-dire transmettre les requêtes du client à d'autres serveurs RADIUS.

Le serveur central (appelé serveur RADIUS) va authentifier ces utilisateurs, et leur autoriser l'accès à telle ou telle ressource. Une autre fonctionnalité importante d'un serveur RADIUS est la comptabilisation des informations concernant les utilisateurs distants ou locaux.

Un utilisateur envoie une requête au NAS pour une demande d'autorisation de connexion à distance. Le NAS transmet la demande au serveur RADIUS. Le serveur RADIUS consulte la base de données d'identification afin de connaître le type de scénario d'identification demandé pour l'utilisateur. Soit le scénario actuel convient, soit une autre méthode d'identification est demandée à l'utilisateur. Le serveur RADIUS retourne ainsi une des quatre réponses suivantes :

ACCEPT : l'identification a réussi.

REJECT : l'identification a échoué.

CHALLENGE : le serveur RADIUS souhaite des informations supplémentaires de la part de l'utilisateur et propose un « défi ».

Une autre réponse est possible : **CHANGE PASSWORD** où le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.

Le protocole RADIUS assurant les échanges entre l'authenticator et le serveur d'authentification est décrit dans les RFC 2865, 2866, 2867, 2868, 2869, 3162. Il utilise le protocole UDP avec les ports 1812 pour l'authentification et 1813 pour la comptabilité. Le format d'un paquet RADIUS est le suivant :

Code	Identifier	Length	Authenticator	Attributes

- Code a une taille de 1 octet et identifie le type de paquet :

1 : **Access-Request**. Du client vers le serveur pour demander l'authentification et l'autorisation d'une connexion.

2 : **Access-Accept.** Du serveur vers le client en réponse à un message Access-Request

3 : **Access-Reject**. Du serveur vers le client en réponse à un message Access-Request pour indiquer que la connexion est refusée.

4: **Accounting-Request**

5: **Accounting-Response**

11: **Access-Challenge**. Du serveur vers le client en réponse à un message Access- Request pour envoyer un défi au client qui devra répondre.

12**: Status-Server**

13: **Status-Client**

255 : **Réservé**

- **Identifier** a une taille de 1 octet et permet d'associer requêtes et réponses.

- **Authenticator** a une taille de 16 octets. Ce champ est utilisé pour authentifier et contrôler l'intégrité des messages RADIUS transmis. Pour simplifier dans une requête il s'agit d'une valeur non prévisible (aléatoire) et globalement unique. Dans la réponse associée c'est une empreinte (hash) de la requête + la réponse + le secret partagé. L'émetteur est alors sûr que la réponse provient du bon interlocuteur (il connaît le secret) et que ni la requête ni la réponse n'ont été altérées.

- **Attributes**. Une liste d'attributs dont la fin est indiquée par la longueur du paquet.

Un attribut a le format suivant :

Type	Length	Value

- Type a une longueur de 1 octet et définit l'attribut. Exemples :

1: **User-Name**

2: **User-Password**

3: **CHAP-Password**

79: **EAP-Message**

80: **Message-Authenticator**

- **Length** a une longueur de 1 octet et indique la taille en incluant les champs type, length et value.

- **Value**. Contient les informations relatives à l'attribut. Le format et la taille de la valeur sont déterminés par les champs type et length. Il y a 5 formats possibles :

Text : 1-253 octets contenant une chaîne de caractères encodés en UTF-8 (ISO 10646)

String : 1-253 octets contenant des données binaires.

Address : 32 bits

Integer : entier 32 bits non signé

Time : nombre de secondes (32 bits) depuis le 01/01/1970 00:00:00

Les messages EAP en provenance du supplicant sont encapsulés par l'authenticator dans un message RADIUS de type EAP-Message (79) pour être transmis au serveur d'authentification.

L'attribut Message-Authenticator est une empreinte MD5 de l'ensemble du message en utilisant le secret partagé comme clé. Cet attribut à été ajouté par la suite, afin d'améliorer la sécurité du protocole RADIUS. Cela permet de se prémunir contre un client pirate effectuant une attaque par dictionnaire contre le serveur RADIUS. Mais ne protège pas contre une attaque par dictionnaire hors ligne par un pirate qui aurait récupéré le défi et la réponse.

On peut légitimement s'interroger si les différentes mesures qui ont été prises pour sécuriser le protocole RADIUS sont suffisantes. Dans le cas du 802.1x il faut cependant très largement tempérer ces craintes par les faits suivants :

- Dans une architecture bien conçue les serveurs RADIUS, les commutateurs sont connectés sur un réseau isolé (VLAN spécifique).

- Avec les méthodes d'authentification EAP-TLS (Extended Authentification Protocol-Transport Layer Security) ou EAP-TTLS (Extended Authentification Protocol-Tunneled Transport Layer Security) le protocole RADIUS ne sert qu'à transporter du TLS et c'est ce dernier qui assure la sécurité de bout en bout.

2. LES PROTOCOLES UTILISES

a. LE 802.1x

Le nouveau standard 802.1x est une réponse au besoin d'authentifier les machines ou les utilisateurs connectés sur un réseau local. Les solutions d'authentification locale basée sur 802.1x fonctionnent bien, à la condition d'avoir un ordinateur relativement récent, c'est-à-dire que le système d'exploitation soit supérieur ou égal à Windows XP. Pour les ordinateurs plus anciens, il n'est pas possible d'effectuer

une authentification de l'utilisateur de manière fiable. Le protocole 802.1x permet d'authentifier les éléments connectés sur le réseau. Le protocole 802.1x définit 3 catégories d'acteurs jouant chacun un rôle différent. Ce sont le requérant *(« supplicant »),* le certificateur *(« authenticator »)* et le serveur d'authentification (« *authentication server* »).

- ***Le supplicant*** est le poste travail demandant à accéder au réseau.
- ***L'authenticator*** est le dispositif, qui dans le cas qui nous concerne, fournit la connexion au réseau : le commutateur. Sur ce dispositif, un port peut avoir 2 états :
    - Non autorisé. Tant que l'authentification du client n'est pas réussie, le port reste dans l'état non autorisé. Seul le trafic entre le requérant (Supplicant) et l'authenticator est permis afin de pouvoir effectuer cette authentification.
    - Autorisé. Après une authentification réussie, le port bascule dans l'état autorisé. La station de travail est alors autorisée à avoir un accès complet au réseau.

  L'authenticator n'effectue aucune authentification, il se contente de jouer le rôle de relais pour transmettre les messages d'authentification en provenance du supplicant vers le serveur d'authentification et vice versa. Si le serveur d'authentification valide la demande alors le port est commuté dans l'état autorisé et alors la station de travail est autorisée à avoir un accès complet au réseau.
- ***Le serveur d'authentification***. Il s'agit d'une machine implémentant un serveur RADIUS. C'est lui qui va authentifier le supplicant. En fonction du résultat de l'authentification ce serveur va envoyer à l'authenticator un message comme quoi ce dernier peut autoriser ou non le port.

Par abus de langage et pour mieux correspondre à la réalité des matériels utilisés, nous appellerons parfois, par la suite, supplicant, authenticator et serveur d'authentification respectivement client ou station de travail, commutateur et serveur.

Il faut noter un certain nombre de points dans cette architecture :

- L'authenticator ne gère pas directement la tache complexe de l'authentification, il la délègue à un serveur. Il se limite à un rôle de relais. C'est donc relativement facile à intégrer à un élément de réseau comme un commutateur.
- Le serveur RADIUS a été initialement conçu pour authentifier des connexions par modem.
- Le serveur RADIUS outre le résultat de l'authentification peut envoyer à l'authenticator des informations supplémentaires comme un numéro de VLAN à utiliser, des règles de filtrage (ACL Access Control List) à appliquer. Il reste qu'il s'agit très souvent d'extensions propriétaires.
- Les méthodes d'authentification ne sont pas fixées par la norme qui ne définit qu'un mécanisme d'échange des messages d'authentification. Cela offre une grande souplesse mais pose aussi de sérieux problèmes d'interopérabilité.
- Si on veut évaluer la qualité globale de l'authentification il ne faut pas s'intéresser uniquement au protocole 802.1x stricto sensu ou à EAP qui reste relativement simple mais aussi aux protocoles RADIUS, TLS et surtout à leurs implémentations. On a affaire à des protocoles complexes dont la validité est dure à montrer et dont la réalisation est nécessairement accompagnée de bogues.
- Pour des raisons de disponibilité le serveur d'authentification doit être redondant, au minimum 2 machines. En effet sans serveur disponible aucune machine ne peut accéder au réseau. Pour équilibrer la charge entres ces différents serveurs, il suffit de ne pas déclarer sur tous les authenticator la même machine comme serveur primaire mais au contraire de les répartir.

A notre connaissance les systèmes d'exploitation implémentant le standard 802.1x (dans le rôle supplicant) sont actuellement :

Windows 2000 SP4, Windows XP, Windows 2003, MacOS X, FreeBSD, OpenBSD, Linux avec open1x.

b. LE PROTOCOLE EAP (Extended Authentification Protocol)

Le protocole utilisé pour assurer l'authentification est EAP (Extensible Authentification Protocol) définit dans le RFC 2284. Ce n'est pas, en soi, un protocole d'authentification, seulement un transport optimisé des informations nécessaires à l'authentification.

Le format d'un paquet est le suivant :

Code	Identifier	Length	Data

- Code a une taille de 1 octet et définit le code du paquet

1: Request

2: Response

3: Success

4: Failure

- Identifier a une taille de 1 octet et permet d'associer les requêtes avec les réponses.

- Length a une taille de 2 octets et définit la longueur totale du paquet incluant code, identifier, length et data.

- Data : zéro ou plusieurs octets de données dont le format dépend du type de paquet.

Les paquets Request et Response ont le format suivant :

Code	Identifier	Length	Type	Type-Data

- Code a une taille de 1 octet et définit le code du paquet

1 : Request

2 : Response

- Identifier a une taille de 1 octet et permet d'associer les requêtes avec les réponses.

- Length a une taille de 2 octets et définit la longueur totale du paquet incluant code, identifier, length, type et type-data.

- Type a une taille de 1 octet et indique le type de la requête ou de la réponse.

- Type-Data : le format varie en fonction du type de requête et de la réponse associée.

Nous préférons pour éviter des confusions ne pas traduire les termes employés dans les différentes normes. Les paquets Success et Failure ont le format suivant :

Code	Identifier	Length

- Code a une taille de 1 octet et définit le code du paquet

  3 : Success

  4 : Failure

- Identifier a une taille de 1 octet et permet d'associer les requêtes avec les réponses.

- Length a une taille de 2 octets et a la valeur 4 puisqu'il n'y a pas de données.

Un certain nombre de types pour les paquets request et response ont été initialement définis dans le RFC mais il est prévu de pouvoir les étendre.

- 1 Identity. Permet de demander l'identité de l'interlocuteur (request) et la fournit (response).

- 2 Notification

- 3 Nak (Response seulement)

- 4 MD5 Challenge

- 5 One Time Password (OTP) (RFC 1938)

- 6 Generic Token Card

- 13 TLS (extension non définie dans le RFC 2284)

Le standard 802.1x définit entre autres choses EAPOL (Extended Authentification Protocol Over LAN) qui est une méthode d'encapsulation d'EAP sur un réseau local. C'est ce qui est utilisé pour le dialogue entre le supplicant et l'authenticator tant que le port reste dans l'état unauthorized. Pour Ethernet on utilise des paquets ayant un type de 0x888e. Pour dialoguer avec l'authenticator on utilise l'adresse 01:80:C2:00:00:03. Elle fait partie d'un groupe d'adresses réservées ne devant jamais être retransmise par un pont ou commutateur. Une trame EAPOL se présente sous la forme suivante en ignorant les adresses MAC :

0x888E Protocol Version Packet Type Packet Body Length Packet Body

- 0x888E est le type Ethernet pour EAPOL

- Protocol Version a une taille de 1 octet et vaut aujourd'hui 1

- Packet Type a une taille de 1 octet et peut prendre les valeurs suivantes :

0 : EAP-Packet. Pour encapsuler un paquet EAP.

1 : EAPOL-Start. Permet au supplicant de démarrer une séquence d'authentification.

2 : EAPOL-Logoff. Lorsque le supplicant envoie ce paquet l'authenticator bascule immédiatement le port dans l'état unauthorized.

3 : EAPOL-Key. Prévu pour la gestion des clés de chiffrement.

4 : EAPOL-Encapsulated-ASF-Alert. Permet de transmettre des alertes (SNMP traps) lorsque le port est dans l'état unauthorized.

- Packet Body Length a une taille de 2 octets et définit la longueur en octets du champ Packet Body.

- Packet Body est présent uniquement pour les types EAP-Packet, EAPOL-Key ou EAPOL-Encapsulated-ASF-Alert et contient les données encapsulées.

Méthodes d'authentification EAP

Il existe différentes méthodes d'authentification pour EAP :

- EAP-MD5. C'est la plus simple. Le client est authentifié par le serveur en utilisant un mécanisme de défi réponse. C'est à dire le serveur envoie une valeur aléatoire (le défi), le client concatène à ce défi le mot de passe et en calcule, en utilisant l'algorithme MD5, une empreinte (« hash ») qu'il renvoie au serveur. Le serveur qui connaît le mot de passe calcule sa propre empreinte, compare les deux et en fonction du résultat valide ou non l'authentification. Une écoute du trafic peut dans le cas d'un mot de passe mal choisi de permettre de le retrouver par une attaque par

dictionnaire ou par force brute. Il n'y a pas d'authentification mutuelle, le serveur n'est pas authentifié par le client.

- EAP-TLS. C'est la plus sûre. Le serveur et le client possèdent chacun leur certificat qui va servir à les authentifier mutuellement. Cela reste relativement contraignant du fait de la nécessité de déployer une IGC (Infrastructure de Gestion de Clés). Rappelons que TLS, la version normalisée par l'IETF de SSL (Secure Socket Layer), est un transport sécurisé (chiffrement, authentification mutuelle, contrôle d'intégrité). C'est lui qui est utilisé de façon sous-jacente par HTTPS, la version sécurisée de HTTP, pour sécuriser le Web. Il faut noter cependant que l'identité de l'utilisateur (paquet EAP-Response/Identity) n'est pas protégée.

- EAP-TTLS (tunneled TLS) utilise TLS comme un tunnel pour échanger des couples attribut-valeur à la manière de RADIUS servant à l'authentification. Pratiquement n'importe quelle méthode d'authentification peut être utilisée.

- PEAP (Protected EAP) est une méthode très semblable dans ses objectifs et voisine dans la réalisation à EAP-TTLS. Elle est développée par Microsoft. Elle se sert d'un tunnel TLS pour faire circuler de l'EAP. On peut alors utiliser toutes les méthodes d'authentification supportées par EAP.

- LEAP (Lightweight EAP) est une méthode propre à Cisco qui repose sur l'utilisation de secrets partagés pour authentifier mutuellement le serveur et le client. Elle n'utilise aucun certificat et est basé sur l'échange de défi et réponse.

Si on veut classer les différentes méthodes, on a de la moins sûre à la plus sûre :

- EAP-MD5.

- LEAP

- EAP-TTLS, PEAP

- EAP-TLS

3. LES GESTIONNAIRES DE DONNEES

Comme il a été dit tant tôt, RADIUS, tout comme VMPS, fonctionne aussi bien avec MySQL qu'avec LDAP.

III. ETUDE COMPRATIVE

a. VMPS vs RADIUS

	VMPS	RADIUS
Exigences	Pour pouvoir fonctionner, il faut que les clients VMPS soient de marque Cisco.	Pas d'exigences particulières
Principe	Identification du supplicant par son adresse MAC	Définition du schéma d'identification suivie de l'authentification de l'utilisateur

Limites et sécurité	Quant à VMPS, un moyen de s'en prendre à une architecture à base de VMPS serait se procurer de l'identité d'une autre station pour entrer sur un VLAN et ainsi bénéficier de tous les privilèges d'un membre du réseau. Un pirate peut trouver une adresse MAC valide sur un VLAN souhaité et y obtenir l'accès. Il peut y arriver, grâce au social engineering, en brute-forçant ou en envoyant une requête sur une adresse MAC Multicast. Le pirate peut aussi injecter diverses requêtes VQP s'il récupère l'adresse MAC du serveur où est stockée la base de données VMPS et fait un ARP hijacking. En utilisant cette méthode, il est aussi possible de spoofer le serveur TFTP et envoyer un fichier de configuration "modifié" au serveur VMPS. Le pirate pourrait alors se connecter aux VLANs de son choix. Sans sniffing, le pirate pourrait récupérer le fichier de configuration VMPS en brute forçant le nom du fichier s'il connaît l'adresse IP du serveur TFTP.	Prenons le cas de RADIUS, nous pouvons citer entre autres limites du serveur : le fait qu'il ait été conçu pour des identifications par modem, sur des liaisons lentes et peu sûres ; c'est la raison du choix du protocole UDP, bien expliquée dans RFC2138. RADIUS base son identification sur le seul principe du couple login/mot de passe ; parfaitement adapté à l'époque (1996), cette notion a dû être adaptée par exemple pour l'identification des terminaux mobiles par leur numéro IMEI ou par leur numéro d'appel (Calling-Station-ID en Radius) sans mot de passe (alors que la RFC interdit le mot de passe vide !). RADIUS assure un transport en clair, seul le mot de passe est chiffré par hachage ; la sécurité toute relative du protocole repose sur le seul shared secret et impose la sécurisation des échanges entre le client et le serveur par sécurité physique ou VPN.
Fonctionnalité	Identification	Authentification + Comptabilisation

Réponse aux besoins	Il répond le mieux aux besoins soulignés précédemment.	La solution qu'offre RADIUS serait difficile gérer dans le réseau.

IV. DEPLOIEMENT DE LA SOLUTION OPTIMALE

Prenant en compte les soucis qui ont été formulés au niveau de la problématique, c'est une solution d'identification qui serait la plus appropriée à notre problème. En effet il a été souligné qu'il serait fastidieux aussi bien pour nous que pour l'administrateur d'attribuer à chaque usager du réseau on login+mot de passe, ajouter à cela, le fait que 95% des équipements du réseau sont de marque Cisco, sont les principales raisons qui nous ont poussé à adopter cette solution à savoir : VMPS avec OpenLDAP.

Il est à noter aussi que le réseau dispose déjà de serveurs DHCP (Dynamic Host Configuration Protocol) pour l'attribution dynamique des adresses IP aux hôtes qui souhaiteraient se connecter au réseau. C'est la raison pour laquelle nous n'en avons pas fait un point à développer dans ce rapport. Mais n'empêche qu'il demeure indispensable à la solution à mettre en œuvre.

Nous allons donc procéder au déploiement de la solution choisie.

1. INSTALLATION ET CONFIGURATION DE LDAP

a. Etape 1 : INSTALLATION

Nous allons installer LDAP sur une machine ayant Debian GNU/Linux kernel 2.6.26-2.686 comme système d'exploitation.

- Les paquets à installer :

Nom	Version	Description
Slapd	2.4.11-1	Le serveur LDAP
Ldap–utils	2.4.11-1	OpenLDAP utilities
Libldap2	2.1.30-13.3	Bibliothèque OpenLDAP
Php5–ldap	5.2.6.dfsg.1-1	Module de LDAP pour php5
PhpLdapAdmin	1.1.0.5-6	Application Web pour l'utilisation du serveur
Dhcp3-server-ldap	3.1.1-6	Serveur DHCP utilisant LDAP comme backend

- Installation à partir du terminal :

Les commandes à saisir (en mode root #):

 # apt-get install slapd

 # apt-get install ldap-utils

 # apt-get install libldap2

 # apt-get install php5-ldap

 # apt-get install phpldapadmin

 # apt-get install dhcp3-server-ldap

Nous pouvons aussi tout installer en une seule ligne de commande :

 # apt-get install slapd ldap-utils libldap2 php5-ldap phpldapadmin dhcp3-server-ldap

b. Etape 2 : CONFIGURATION

Le fichier de configuration de LDAP situé dans */etc/ldap/slapd.conf.* Le premier réflexe que nous avons eu avant de passer à la configuration du serveur a été de faire une sauvegarde de copie des fichiers de configuration du serveur qui va nous permettre de ne pas perdre le fichier original en cas d'erreur de configuration:

- *cp /etc/ldap/slapd.conf /etc/ldap/slapd.conf.orig*

Voici le contenu du fichier slapd.conf (dans ces fichiers, ce qui est précédés par un '#' est un commentaire, donc pas interprété par le serveur):

```
# This is the main slapd configuration file. See slapd.conf(5) for more
# info on the configuration options.
########################################################
# Global Directives:
# Features to permit
#allow bind_v2
# Schema and objectClass definitions
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema
#schemacheck on
# Where the pid file is put. The init.d script
# will not stop the server if you change this.
pidfile         /var/run/slapd/slapd.pid
# List of arguments that were passed to the server
argsfile        /var/run/slapd/slapd.args
# Read slapd.conf(5) for possible values
loglevel        none
# Where the dynamically loaded modules are stored
modulepath      /usr/lib/ldap
moduleload      back_hdb
# The maximum number of entries that is returned for a search operation
sizelimit 500
# The tool-threads parameter sets the actual amount of cpu's that is used
# for indexing.
tool-threads 1
########################################################
# Specific Backend Directives for hdb:
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
backend                 hdb
########################################################
# Specific Backend Directives for 'other':
# Backend specific directives apply to this backend until another
# 'backend' directive occurs
#backend                <other>
########################################################
```

```
# Specific Directives for database #1, of type hdb:
# Database specific directives apply to this databasse until another
# 'database' directive occurs
database        hdb
# The base of your directory in database #1
suffix          "dc=ucad,dc=sn"
# rootdn directive for specifying a superuser on the database. This is needed
# for syncrepl.
rootdn          "cn=admin,dc=ucad,dc=sn"
rootpw          secret
# Where the database file are physically stored for database #1
directory       "/var/lib/ldap"
# The dbconfig settings are used to generate a DB_CONFIG file the first
# time slapd starts.  They do NOT override existing an existing DB_CONFIG
# file.  You should therefore change these settings in DB_CONFIG directly
# or remove DB_CONFIG and restart slapd for changes to take effect.
# For the Debian package we use 2MB as default but be sure to update this
# value if you have plenty of RAM
dbconfig set_cachesize 0 2097152 0
# Sven Hartge reported that he had to set this value incredibly high
# to get slapd running at all. See http://bugs.debian.org/303057 for more
# information.
# Number of objects that can be locked at the same time.
dbconfig set_lk_max_objects 1500
# Number of locks (both requested and granted)
dbconfig set_lk_max_locks 1500
# Number of lockers
dbconfig set_lk_max_lockers 1500
# Indexing options for database #1
index           objectClass eq
# Save the time that the entry gets modified, for database #1
lastmod         on
# Checkpoint the BerkeleyDB database periodically in case of system
# failure and to speed slapd shutdown.
checkpoint      512 30
# Where to store the replica logs for database #1
# replogfile    /var/lib/ldap/replog
# The userPassword by default can be changed
# by the entry owning it if they are authenticated.
# Others should not be able to see it, except the
# admin entry below
```

```
# These access lines apply to database #1 only
access to attrs=userPassword,shadowLastChange
        by dn="cn=admin,dc=ucad,dc=sn" write
        by anonymous auth
        by self write
        by * none
# Ensure read access to the base for things like
# supportedSASLMechanisms.  Without this you may
# have problems with SASL not knowing what
# mechanisms are available and the like.
# Note that this is covered by the 'access to *'
# ACL below too but if you change that as people
# are wont to do you'll still need this if you
# want SASL (and possible other things) to work
# happily.
access to dn.base="" by * read
# The admin dn has full write access, everyone else
# can read everything.
access to *
        by dn="cn=admin,dc=ucad,dc=sn" write
        by * read
# For Netscape Roaming support, each user gets a roaming
# profile for which they have write access to
#access to dn=".*,ou=Roaming,o=morsnet"
#       by dn="cn=admin,dc=ucad,dc=sn" write
#       by dnattr=owner write
#######################################################
# Specific Directives for database #2, of type 'other' (can be hdb too):
# Database specific directives apply to this databasse until another
# 'database' directive occurs
#database       <other>
# The base of your directory for database #2
#suffix         "dc=debian,dc=org"
```

Etape suivante, faire communiqué les serveurs LDAP et DHCP.

c. Etape 3 : CONFIGURATION LDAP-DHCP

La configuration que nous allons faire dans cette nous permettra de faire communiquer le serveur LDAP et le serveur DHCP. Cela va nous permettre d'enregistrer dynamiquement, dans l'annuaire, toute machine laquelle l'accès a été autorisé par le VMPS.

Cependant LDAP avait quelques manquements. Les schémas proposés par OpenLDAP ne permettent pas de gérer les services DHCP. C'est la raison pour laquelle nous avons installé le module network du MDS (**M**andriva **D**irectory **S**erver) pour les combler. Pour l'installation de ces paquets, nous avons ajouté ce lien dans le fichier */etc/apt/source.list* contenant la liste des liens d'où le système tire ses paquets pour de mise à jour :

```
# Mandriva Directory Server
deb http://mds.mandriva.org/pub/mds/debian etch main
```

ensuite, nous avons lancer le téléchargement des paquets via le terminal:

 # aptitude install mmc-agent mmc-web-base mmc-web-network python-mmc-network slapd ldap-utils

Puis copier ces schémas dans le répertoir de schémas du serveur LDAP:

#cp /usr/share/doc/python-mmc-base/contrib/ldap/mmc.schema /etc/ldap/schema/

#cp /usr/share/doc/python-mmc-base/contrib/ldap/dnszone.schema /etc/ldap/schema/

#cp /usr/share/doc/python-mmc-base/contrib/ldap/dhcp.schema /etc/ldap/schema/

Après cela fait, il faut ajouter ces schémas dans le fichier de configuration du serveur LDAP situé dans /etc/ldap/slapd.conf :

```
include         /etc/ldap/schema/mmc.schema
include         /etc/ldap/schema/dnszone.schema
include         /etc/ldap/schema/dhcp.schema
```

Maintenant pour que ces modifications puissent prendre effet, nous redémarrons le service :

 # /etc/init.d/slapd restart.

Le serveur LDAP est maintenant prêt pour fournir les différents champs nécessaires aux serveurs DNS et DHCP, il reste maintenant à configurer le plugin Network du MDS. Toute la configuration se fait dans */etc/mmc/plugins/network.ini* :

```
[main]
disable = 0

[dhcp]
dn = ou=DHCP,dc=ucad,dc=sn
```

```
pidfile = /var/run/dhcpd.pid
init = /etc/init.d/dhcp3-server
logfile = /var/log/syslog
leases = /var/lib/dhcp3/dhcpd.leases

[dns]
dn = ou=DNS,dc=example,dc=org
pidfile = /var/run/bind/run/named.pid
init = /etc/init.d/bind9
logfile = /var/log/daemon.log
bindroot = /etc/bind/
bindgroup = bind
# dnsreader = DNS Reader
# dnsreaderpassword = DNSReaderPassword
```

puis redémarrer le serveur pour que l'agent-mmc prennent ces modification en compte :

 # /etc/init.d/mmc-agent restart

Après cela, nous allons créer un fichier nommé *dhcpd.conf.ldap* dans le répertoire */etc/dhcp3/* contenant ceci :

```
# Le serveur LDAP
 ldap-server "localhost";
# Le port utilisé
 ldap-port 389;
# Le DN de l'utilisateur que l'on utiliser pour aller lire l'arbre LDAP
 ldap-username "cn=dhcpadmin,ou=dhcp,dc=ucad,dc=sn";
# Son password
 ldap-password "dhcppassword";
# Le base DN de la branche où toutes les informations sur le DHCP seront stockées
 ldap-base-dn "ou=dhcp,dc=ucad,dc=sn";
#La méthode: si on met dynamique, le serveur DHCP va interroger l'arbre LDAP à
#chaque fois
#Il n'y a donc pas besoin de redémarrer quand on ajoute une entrée
ldap-method dynamic;
# Le fichier de log où l'on voit la configuration que le serveur DHCP construit à
#partir des informations qu'il va lire dans le LDAP
 ldap-debug-file "/var/log/dhcp-ldap-startup.log";
```

C'est ce fichier qui donne les directives que doit suivre le serveur DHCP pour pouvoir communiqué avec LDAP. Il nous reste maintenant à l'inclure dans le fichier de configuration de DHCP */etc/dhcp3/dhcpd.conf* :

```
include "/etc/dhcp3/dhcpd.conf.ldap";
```

Nous retournons dans le fichier de configuration de LDAP, *slapd.conf* pour y ajouter l'indexation des paramètres en rapport avec le DHCP, fournis par *dhcp.schema* :

```
index        dhcpHWAddress                    eq
index        dhcpClassData                    eq
```

et un utilisateur spécial pour parcourir cette branche de l'arbre :

```
# DHCP admin ACL
access to dn.subtree="ou=dhcp, dc=ucad,dc=sn"
            by dn.regex="cn=dhcpadmin,ou=dhcp,dc=ucad,dc=sn" write
            by * auth
```

Nous allons procéder à des ajouts d'informations dans l'annuaire LDAP pour que puisse fonctionner correctement notre serveur DHCP. Ce sera par le biais des fichiers LDIF que cela va se faire. Nous allons d'abord créer une branche dans l'arbre pour y mettre toutes les configurations du serveur DHCP. Voici le contenu du LDIF :

```
dn: ou=dhcp,dc=ucad,dc=sn
objectClass: top
objectClass: organizationalUnit
ou: dhcp
description: All informations about DHCP
```

L'ajout se fait grâce à la commande : *ldapadd -x -D "cn=admin,dc=ucad,dc=sn" -w secret -f nom-fichier.ldif* ou bien nous importons le fichier grâce à l'application web phpldapadmin.

L'ajout de l'utilisateur *dhcpadmin* que nous avons précédemment mentionné dans le fichier *slapd.conf :*

```
dn: cn=dhcpadmin,ou=dhcp,dc=ucad,dc=sn
objectClass: top
objectClass: person
userPassword: secret
cn: dhcpadmin
sn: dhcpadmin user
```

Nous allons maintenant créer un objet *dhcpService* à travers lequel nous allons renseigner les options de base de notre service DHCP ainsi que le serveur :

```
# La définition de notre objet
dn: cn=DHCPConfig,ou=DHCP,dc=ucad,dc=sn
cn: DHCPConfig
dhcpPrimaryDN: cn=mass-pc,ou=DHCP,dc=ucad,dc=sn
objectClass: top
objectClass: dhcpService
```

Définir le serveur DHCP primaire :

```
dn: cn=mass-pc,ou=DHCP,dc=ucad,dc=sn
cn: mass-pc
dhcpServiceDN: cn=DHCPConfig,ou=DHCP,dc=ucad,dc=sn
```

objectClass: top
objectClass: dhcpServer

et enfin ses options de base lui permettant de faire son travail:

dn: cn=192.168.0.0,cn=DHCPConfig,ou=DHCP,dc=ucad,dc=sn
cn: 192.168.0.0
dhcpNetMask: 24
dhcpOption: domain-name-servers 192.168.0.1
dhcpOption: subnet-mask 255.255.255.0
dhcpOption: domain-name "ucad.sn"
dhcpRange: 192.168.0.10 192.168.0.254
dhcpStatements: default-lease-time 28800
dhcpStatements: max-lease-time 28800
objectClass: dhcpOptions
objectClass: dhcpSubnet
objectClass: top

2. INSTALLATION ET CONFIGURATION DE VMPS

Avec le serveur VMPS, nous avons deux options d'implémentation :

Option 1 : Prendre un commutateur des séries Catalyst 6000, 6500 ou catalyst 5000, 5500 et le configurer comme serveur. Option que nous n'avons pas pu mettre en pratique car le matériel n'était pas sur place,

Option 2 : Utiliser OpenVMPS. Dans ce cas, nous allons prendre une machine linux comme serveur.

a. Etape 1 : INSTALLATION

Ici nous faisons allusion à l'option 2 qui consiste à utiliser OpenVMPS. Il sera installé sur la même machine où nous avons installé notre serveur LDAP (Debian GNU/Linux kernel 2.6.26-2.686). Le package à installer s'appelle ***vmpsd-1.3.tar.***

Nous lançons ces commandes au terminal :

 # tar xzf vmpsd-1.3.tar.gz

 # cd vmpsd

 # ./configure && make && make install

Après installation, le fichier vlan.db se trouvant dans le répertoire vmpsd/ est un exemple de configuration de serveur VMPS. Etant donné que le fichier de configuration de VMPS doit, par défaut, se situé dans le répertoire /etc/, nous avons décidé de faire une copie de vlan.db que nous allons nommer vmps.db dans ce même répertoire :

 # cp /vlan.db /etc/vmps.db

Maintenant nous allons passer à sa configuration.

b. Etape 2 : CONFIGURATION

Nous avons apporté des modifications sur la copie que nous avons faite du fichier vlan.db pour l'approprier à nos besoins :

Voici la configuration que nous avons faite du serveur :

```
! Le domaine VMPS qui doit obligatoirement être le même que le domaine VTP
vmps domain ucad.sn
! Le mode de sécurité du serveur { open | secure }
vmps mode open
! Définition d'un Vlan 'poubelle' pour les adresses MAC non reconnus dans la base
vmps fallback --NONE--
!Autorisation ou non des requêtes non-accompagnées du domaine VTP
vmps no-domain-req allow
```

Dans le cas de l'utilisation d'OpenVMPS la base de données contenant les associations adresse MAC / VLAN est incluse dans le fichier. Pas besoin donc d-un serveur TFTP ou son équivalent pour y télécharger cette base.

Voici comment sont disposées ses informations dans le fichier :

```
!
!MAC Addresses
!
vmps-mac-addrs
!
```

```
! address <addr> vlan-name <vlan_name>
!
! netreg extension - default vlan
address 00e0.4cd0.225c vlan-name --DEFAULT--
! disabled - no access
address 00e0.4cd0.225c vlan-name --NONE--
! vlan TEST restricted
address 00e0.4cd0.225c vlan-name  100
! vlan TEST1 unrestricted
address 0010.a49f.30e4 vlan-name  200
!
!....
```

Cette partie du fichier qui va suivre permet de spécifier au serveur s'il y a des commutateurs clients mis à la disposition d'un groupe particulier ainsi que les ports de ses commutateurs.

```
!
!Port Groups
!
!vmps-port-group <group-name>
! default-vlan <vlan-name>
! fallback-vlan <vlan-name>
! device <device-id> { port <port-name> | all-ports }
!vmps-port-group groupe100
!  default-vlan100
!  fallback-vlan VLAN2
!  device 192.168.1.20 port fa0/1
!  device 192.168.1.20 port fa0/2
!  device 192.168.1.20 port fa0/9
! device 192.168.1.20 all-ports
```

Mais comme vous pouvez le constater, toute cette partie est commentée car nous n'avions pas à notre disposition beaucoup de ressources. Nous allons maintenant configurer notre client VMPS.

hostname DI-Com

```
!
vmps server 192.168.1.15 primary
```

```
!vmps server 147.173.58.201
!vmps server 147.173.58.202
!
interface FastEthernet0/1
switchport access vlan dynamic
switchport mode access
no ip address
spanning-tree portfast
!
end
```

Retournons au terminal pour demarrer le service :

D'abord il faut arrêter les processus :

 # killall

ensuite demarrer le serveur :

#./vmpsd –a 192.168.1.15 –f /etc/vmps.db

Enfin pour vérifier si le service est bien démarré :

 # tail –f /var /log/messages

Si toute la configuration est correcte, c'est ce message qui s'affichera :

VMPS server waiting for request (En attente des requêtes !!).

Ainsi nous venons de permettre au serveur ainsi qu'au commutateur client de pouvoir s'échanger de paquets nés des requêtes envoyées par les stations de travail.

CONCLUSION

Le stage que nous venons d'effectuer à la Direction Informatique de l'UCAD nous a été d'un grand apport dans la mesure où il nous a permis non seulement de faire face à quelques réalités du monde professionnel, mais aussi d'étendre nos connaissances en matière de sécurité réseau, manipulation d'équipements réseaux (serveurs, commutateurs,…) et surtout de l'utilité du travail de groupe.

Cependant, compte tenu de la brièveté du stage, nous n'avons pu implémenter toutes les fonctionnalités que nous voulions, c'est la raison pour laquelle, la configuration de base porrait être améliorée plus tard par l'équipe de la direction informatique.

LEXIQUE

Un **VLAN** (Virtual Local Area Network) est un réseau local regroupant un ensemble de machines de façon logique et non physique. En effet dans un réseau local la communication entre les différentes machines est régie par l'architecture physique. Grâce aux réseaux virtuels (VLANs) il est possible de s'affranchir des limitations de l'architecture physique (contraintes géographiques, contraintes d'adressage, ...) en définissant une segmentation logique (logicielle) basée sur un regroupement de machines grâce à des critères (adresses MAC, numéros de port, protocole, etc.).

Un **protocole** est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon ce que l'on attend de la communication.

Une **adresse MAC** (Media Access Control address) est un identifiant physique stocké dans une carte réseau ou une interface réseau similaire et utilisé pour attribuer mondialement une adresse unique au niveau de la couche de liaison (couche 2 du modèle OSI). C'est la partie inférieure de celle-ci (sous-couche d'accès au média – Media Access Control) qui s'occupe d'insérer et de traiter ces adresses au sein des trames transmises.

Un **serveur** est un ordinateur et un logiciel dont le rôle est de répondre automatiquement à des demandes envoyées par des clients - ordinateur et logiciel - via le réseau.

UDP (User Datagram Protocol, en français **protocole de datagramme utilisateur**) est un des principaux protocoles de télécommunication utilisés par Internet. Il fait partie de la couche transport de la pile de protocole TCP/IP : dans l'adaptation approximative de cette dernière au modèle OSI, il appartiendrait à la couche 4, comme TCP. Il est détaillé dans la RFC 768.

TCP (Transmission Control Protocol, soit en français: Protocole de Contrôle de Transmission) est un des principaux protocoles de la couche transport du modèle TCP/IP. Il permet, au niveau des applications, de gérer les données en provenance (ou à destination) de la couche inférieure du modèle.

Un lien **TRUNK** est un lien qui permet de faire transiter plusieurs VLAN sur un seul lien physique.
Sur un LAN avec plusieurs VLAN sur plusieurs commutateurs on peut donc faire circuler ces VLANs sur tous les commutateurs avec un seul lien entre deux commutateurs (sinon il faut un lien par VLAN).

Le **File Transfer Protocol** (*protocole de transfert de fichiers*), ou **FTP**, est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.

SSL (Secure Sockets Layer) est un protocole de sécurisation des échanges sur Internet, devenu Transport Layer Security (TLS) en 2001.

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. Le cadre découple les mécanismes d'authentification des protocoles d'application, permettant en théorie à n'importe quel mécanisme d'authentification pris en charge par SASL d'être employé à partir de n'importe quel protocole d'application capable d'utiliser SASL.

Extensible Authentication Protocol (EAP) est un mécanisme d'identification universel, fréquemment utilisé dans les réseaux sans fil (ex : de type Wi-Fi) et les liaisons point à point.