UNIVERSITE KASDI MERBAH OUARGLA

Faculté des Sciences et de La Technologie et Sciences de La Matière

Département Mathématique Et Informatique

Mémoire
MASTER ACADEMIQUE

Domaine : Mathématique et Informatique

Filière : Informatique Académique

Spécialité : Informatique fondamental

Présenté par : MIHI Abdel Hakim

TERBAGOU Amina

Thème

Authentification unifiée Pour l'accès aux services web de l'université

Soutenu publiquement

Le : 06 /2013

Devant le Jury :

Mr. KAFI Med Redouane	Rapporteur	UQM Ouargla
Mr. *DJEDIAI Hmida*	President	UQM Ouargla
Mr. BELOUAAR Hocine	Examinateur	UQM Ouargla

Année Universitaire : 2012/2013

Remerciements

Louange à ﷲ Seigneur de l'univers

Nous remercions respectueusement notre promoteur, Mr. KAFI Med Redouane, pour tous les conseils et encouragements dont nous avons bénéficiés tout au long de ce travail.

Nos respects et notre gratitude vont également aux membres du jury : Mr DJEDIAI Hmida et Mr BELOUAAR Hocine qui nous ont fait l'honneur de juger ce travail.

Nous remercions également toute l'équipe du Master informatique, nos enseignants, merci aussi à l'équipe de Centre des Systèmes et Réseaux Informatique et Télé enseignement et Visioconférence, Université **KASDI MERBAH OUARGLA** *pour tous les conseils et encouragements dont nous avons bénéficiés tout au long de ce travail.*

Enfin, un grand Merci à tous ceux qui m'ont soutenu de près ou de loin. Merci à mes camarades de la promotion 2012/2013

Hakim et Amina

Dédicaces

Grace à Dieu voilà notre travail terminé et il est temps pour moi de partager ma joie avec tous ceux qui m'ont soutenu et encouragé.

Et à travers cette modeste thèse je tiens à présenter mes sincères dédicaces à

Mon cher père (رحمة الله عليه) qui m'a confié de suivre mes études et m'a donné l'espoir de réussir

Ma mère qui a consacré sa vie à notre éducation et à faire notre bonheur et qui nous encourage toujours d'achever nos études tout en espérant de voir les fruits de ses sacrifices.

A mes chères sœurs et à mes chers frères

A mes Oncles, Tantes et cousins et alliés de la famille.

A l'ensemble des amis que j'ai connu pendant mes études et à ceux qui m'ont prodigué leurs vifs conseils, encouragements et témoigné de leur amitié.

A mon binôme Amina et sa famille.

Hakim

Dédicaces

Grace à Dieu voilà notre travail terminé et il est temps pour moi de partager ma joie avec tous ceux qui m'ont soutenu et encouragé.

Et à travers cette modeste thèse je tiens à présenter mes sincères dédicaces à mes Parents qui ont consacré leurs vie à notre éducation et à faire notre bonheur et qui m'encouragent toujours pour achever mes études tout en espérant de voir les fruits de leurs sacrifices.

A mes chères sœurs et mes chers frères

A mes Oncles, Tantes et cousins et alliés de la famille.

A l'ensemble des amis que j'ai connu pendant mes études et à ceux qui m'ont prodigué leurs vifs conseils, encouragements et témoigné de leur amitié.

A mon binôme Hakim et sa famille.

Amina

Sommaire

Sommaire

Liste des figures

Liste des figures

\

Listes des tableaux

Résumé

Notre travail consiste à faire une contribution à l'unification de l'authentification, pour les différentes plates formes de l'université (Zimbra, Joomla et Dokeos).

Ces plateformes se connecte entre elles pour les bénéfices communs, mais ceci pose un problème pour les utilisateurs et les administrateurs, pour crée et sauvegardé, ou modifiée les noms d'utilisateur/mots de passes de chaque plate forme. Donc le besoin d'unifier l'authentification de toutes les plates formes de notre université est primordiale, et offre un confort et souplesse d'utilisation.

Le concept qu'on a choisi comme solution est d'unifiée l'authentification de touts les plateformes dans un annuaire LDAP.

Mot clés : Authentification, Zimbra, Joomla, Dokeos, Plateforme, Mot de passe, Authentification forte.

Abstract

Our work is to make a contribution to the unification of authentication for different platforms university (Zimbra, Joomla and Dokeos).

These platforms connect them to the common benefits, but this poses a problem for users and administrators to create and saved, or modified the username / passwords for each platform names. So the need to unify the authentication of all platforms of our university is paramount, and offers comfort and flexibility.

The concept that a solution is chosen as unified wholes authentication platforms in an LDAP directory.

Keyword: Authentication, Zimbra, Joomla, Dokeos platform, Password, strong authentication.

ملخص

مهمتنا هي تقديم مساهمة لتوحيد التوثيق لمختلف منصات جامعة Zimbra, Joomla, Dokeos

هذه المنصات تربطهم المنافع المشتركة، ولكن هذا يطرح مشكلة بالنسبة للمستخدمين والإداريين لخلق وحفظها، أو تعديل اسم المستخدم / كلمة السر لكل أسماء منصة. وبالتالي فإن الحاجة لتوحيد المصادقة من جميع المنابر من جامعتنا أمر بالغ .الأهمية، ويوفر الراحة والمرونة

مفهوم أن يتم اختيار الحل كما أجمعين موحد منصات المصادقة في دليل LDAP

الكلمات المفتاحية : مصادقة ، منصة، كلمة المرور، مصادقة قوية Zimbra, Joomla, Dokeos

Introduction General

La sécurité informatique est devenue aujourd'hui vitale dans la gestion des réseaux d'entreprise, ainsi que pour les particuliers toujours plus nombreux à se connecter sur internet. La transmission d'informations sensibles, et le souci d'assurer la confidentialité de celles-ci est devenue un point primordial dans la mise en place des réseaux informatiques.
Un des concepts pour atteindre la sécurité et de l'améliorer c'est ***l'authentification.***

Cette solution constitue une sécurité relativement fiable lorsqu'elle est utilisée avec de fortes méthodes, confidentialité assurée, fichier protégé, mais cette authentification lorsqu'on utilise de nombreux applications web et plusieurs stations différentes, l'utilisateur rencontre des problèmes pour crée plusieurs compte dans des différents plateformes, mémoriser leur mots de passe, ou chercher un identifiant qui peut être pas facile de s'en rappeler alors l'utilisateur inscrit ces codes secrets dans son agenda papier, le note sur des Post-it qu'il colle autour de leur écran ou plus simple, laisse leur connexion ouverte lorsqu'il quitte leur poste de travail. Et ce, afin de ne pas avoir à répéter le rituel quotidien de l'accès sécurisé à leur application

Problématique :
Notre environnement universitaire a différentes plateformes qui sont utilisées par les enseignants, étudiants et staff administratif pour se connecter entre eux et partagées les informations et les événements tous ceci pour une collaboration d'intérêt commun, mais des problèmes existent toujours :

- Crée un compte pour chaque plateforme.
- Mémoriser chaque identifiant et chaque mot de passe
- Multiples identifiants qui peuvent être un casse-tête pour les mémoriser et les réécrire à chaque session

Objectifs :

Dans ce travail on veut présenter globalement une solution pour ces problèmes qui touche notre environnement universitaire surtout qu'on a des plateformes sensibles connectées entre elles. Donc l'objectif de notre travail et de faire une ***authentification unifiée*** adaptée à l'environnement universitaire pour cette solution on utilise deux méthodes la première méthode un serveur LDAP (Open LDAP) comme un annuaire pour les comptes d'utilisateurs qui utilise aussi zimbra et Dokeos et joomla alors l'accès sera facile avec une authentification sécurisé et la deuxième c'est qu'on utilise un LDAP Zimbra qui est bien configuré avec Zimbra ça sera un annuaire avec bien sûr les autre plateformes et cette solution est bien illustrer dans ce qui suit .

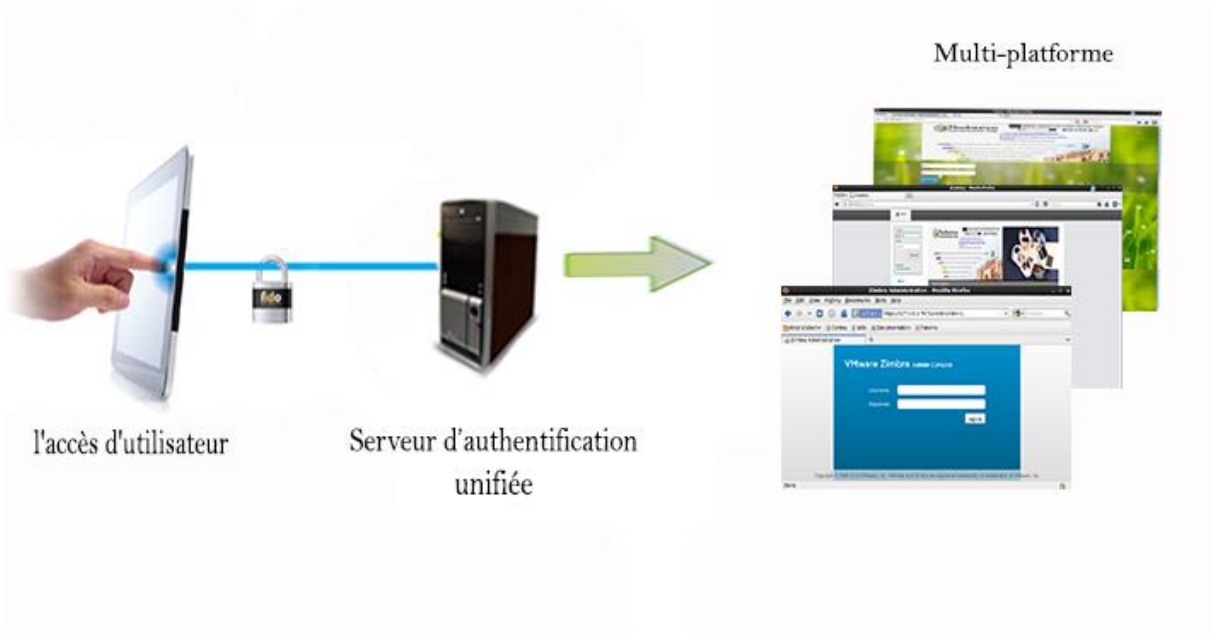


Figure 01 : présentation de notre objectifs de travail l'authentification unifiée

Organisation du mémoire :

Notre mémoire est organisé en 4 chapitres :

Chapitre I : Dans ce chapitre on a représenté le concept de sécurité et l'aspect de l'authentification qui est important dans notre application web, on a montré aussi leurs techniques, les avantages et les inconvénients et les différentes méthodes d'authentification

Chapitre II : Dans ce chapitre on a parlé des techniques d'authentification le SSO (singl- sign on)et le CAS (Central Authentication Service)…

Chapitre III : Dans ce chapitre on a représenté les outils qu'on a utilisés dans notre travail : le système CentOS, Open LDAP, Zimbra, Joomla et Dokeos.

Chapitre IV :dans ce chapitre on a présenté la conception de notre travail et l'implémentation, la mise en œuvre de LDAP avec les autres plateformes.

Chapitre I : L’authentification

1. Introduction

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou à leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information.

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

2. Sécurité informatique

La sécurité informatique est l'ensemble des techniques qui assurent que les ressources du système d'information (matérielles ou logicielles) d'une organisation sont utilisées uniquement dans le cadre où il est prévu qu'elles le soient.

Les techniques de la sécurité informatique se divisent comme suit :
- analyse des risques .
- politique de sécurité .
- techniques de sécurisation.

2.1. Analyse des risques

Plus aucune entreprise ne peut se passer de l'outil informatique, d'où la nécessité d'en assurer la sécurité, et de la protéger contre les risques liés à l'informatique. Or, comme on ne se protège efficacement que contre les risques qu'on connaît, il importe de mesurer ces risques, en fonction de la probabilité ou de la fréquence de leur apparition et de leurs effets possibles. Chaque organisation a intérêt à évaluer, même grossièrement, les risques qu'elle court et les protections raisonnables à mettre en œuvre. Les risques et les techniques de sécurisation seront évalués en fonction de leurs coûts respectifs.

2.2. Politique de sécurité

À la lumière des résultats de l'analyse des risques, la politique de sécurité :

- Définit le cadre d'utilisation des ressources du système d'information .
- Identifie les techniques de sécurisation à mettre en œuvre dans les différents services de l'organisation .
- Sensibilise les utilisateurs à la sécurité informatique.

2.3. Techniques de sécurisation

Elles assurent la disponibilité (les services et les informations doivent être accessibles aux personnes autorisées quand elles en ont besoin et dans les délais requis), l'intégrité (les services et les informations ne peuvent être modifiés que par les personnes autorisées), et la confidentialité (l'information est accessible uniquement à ceux qui y ont droit). Les techniques de sécurisation d'un système incluent :

- Audit de vulnérabilités, essais de pénétration.
- Sécurité des données: chiffrement, authentification, contrôle d'accès.
- Sécurité du réseau: pare-feu.
- Surveillance des informations de sécurité.
- Éducation des utilisateurs .
- Plan de reprise des activités.

3. Authentification

Une solution d'authentification met en jeu une chaîne d'acteurs et de processus allant de la création de l'élément authentifiant jusqu'à son contrôle.

L'authentification vise à acquérir un certain niveau de certitude quant à l'authenticité d'un produit afin de produire des éléments de preuve. Cette certitude s'acquiert en vérifiant l'authenticité des éléments authentifiants du produit choisis parmi l'offre de dispositifs. De ce fait, le mode de génération, de réalisation, de marquage (peu en importe la forme) et de capture de ces éléments authentifiants doit être sécurisé ainsi que leur association avec le produit à protéger.

Une solution d'authentification comprend :

– la création du ou des éléments authentifiants.

– l'intégration ou l'association au produit du ou des éléments authentifiants.

– les outils de contrôle.

– la vérification par un contrôleur formé.
– la capacité à mesurer l'efficacité du dispositif et des protocoles associés.
– la capacité à gérer des crises.

Dans un serveur, un processus de contrôle valide l'identité et après authentification, donne l'accès aux données, applications, bases de données, fichiers ou sites Internet. Dans le cas contraire, l'accès est refusé.

L'authentification peut se faire de multiples manières, et notamment par la vérification de :

« Ce que je sais », un mot de passe par exemple,

« Ce que je sais faire », une signature manuscrite sur écran tactile/digital (de type PDA),

« Ce que je suis », une caractéristique physique comme une empreinte digitale,

« Ce que je possède », une carte à puce par exemple.

Le choix de telle ou telle technique dépend en grande partie de l'usage que l'on souhaite en faire : authentification de l'expéditeur d'un email, authentification d'un utilisateur qui se connecte à distance, authentification d'un administrateur au système, authentification des parties lors d'une transaction de **B2B** (Business to Business), etc…

La combinaison de plusieurs de ces méthodes (aussi appelées facteurs d'authentification) permet de renforcer le processus d'authentification, on parle alors d'authentification forte.

Les techniques d'authentification les plus usitées sont, de loin, les mots de passe mais aussi, de plus en plus, les Certificats de clés publiques**.**

On peut considérer que l'authentification forte est une des fondations essentielles pour garantir :

- L'autorisation ou contrôle d'accès (qui peut y avoir accès)
- La confidentialité (qui peut le voir)
- L'intégrité (qui peut le modifier)
- La traçabilité (qui l'a fait).***[01]***

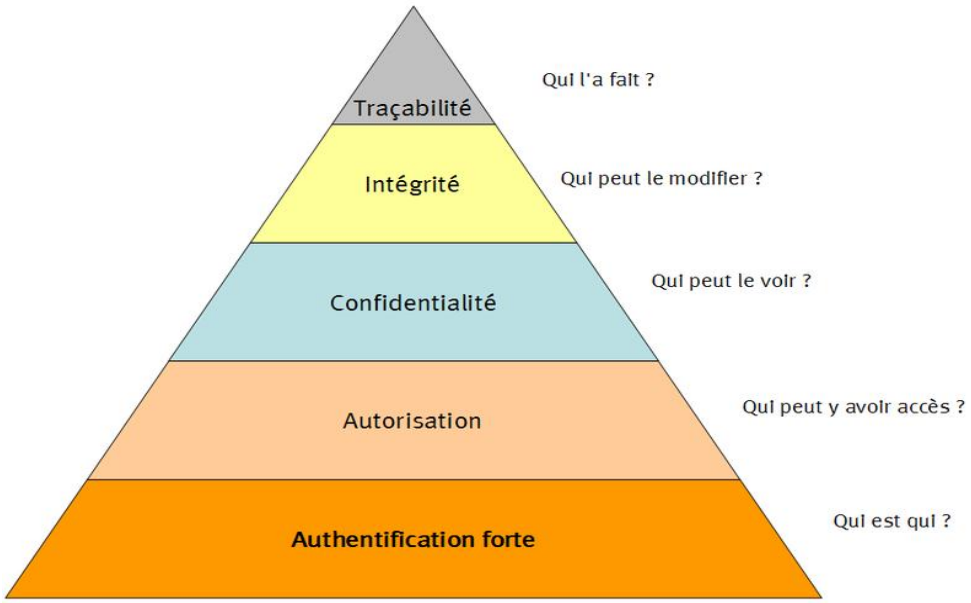


Figure 1: Représentation de l'authentification forte sous forme pyramidale

4. Méthodes d'authentification

4.1. L'identifiant et le mot de passe

L'identifiant et le mot de passe sont le couple d'authentification le plus connu. Simple, robuste, voire même rustique, son plus gros défaut est que le niveau de sécurité dépend directement de la complexité du mot de passe. Des mots de passes simples sont faibles, et des mots de passes trop complexes conduisent les utilisateurs à mettre en œuvre des stratégies de contournement pour les gérer : Post-it, liste dans un fichier Excel ou dans le Smartphone,…

4.2. L'identifiant et le mot de passe OTP (One-Time Password)

L'OTP permet de sécuriser l'utilisation du mot de passe sur le réseau. En effet avec un système OTP, l'utilisateur possède un calculateur spécialisé qui lui fournit à la demande un mot de passe. Ce mot de passe est valide pendant une durée limitée seulement, et pour une seule utilisation. Cette solution est en général mise en œuvre pour le processus d'authentification initiale pour les accès externes via IP/VPN[1] .

[1] VPN :**Virtual Private Network**, réseau privé virtuel en anglais, une connexion inter-réseau permettant de relier deux réseaux locaux différents par un protocole de tunnel.

4.2.1 Mise en œuvre et exploitation

Cette solution suppose en générale la mise en œuvre d'un ou plusieurs serveurs spécifiques d'authentification accessibles. Chaque utilisateur doit posséder une calculatrice spécifique et le mot de passe associé. Il faut donc mettre en place les procédures de gestion des demandes utilisateurs suite à la perte ou l'oubli d'une calculatrice ou à l'oubli d'un mot de passe.

4.3. Les certificats PKI sur carte à puce ou clef USB

Les certificats X.509 mettent en œuvre une technologie avancée de chiffrement qui permet de chiffrer ou signer des messages sans avoir à partager de secret. L'identifiant est un certificat public qui est signé et donc garanti par une autorité de certification reconnue. L'utilisateur doit fournir un secret pour pouvoir utiliser les différents éléments cryptographiques : « le code PIN de sa carte ou de sa clef USB».

La mise en œuvre d'une solution à base de carte à puce et de certificat suppose l'agrégation de plusieurs composants

- La carte avec son lecteur ainsi que le code logiciel associé qui doit être installé surle poste de travail.
- L'infrastructure de certificat X.509 doit fournir les différents composants d'une infrastructure PKI : l'Autorité de Certification et l'Autorité d'Enregistrement.
- Le CMS (Card Management System) qui va gérer l'attribution des cartes.
- Le module d'authentification.
- Le serveur d'authentification.

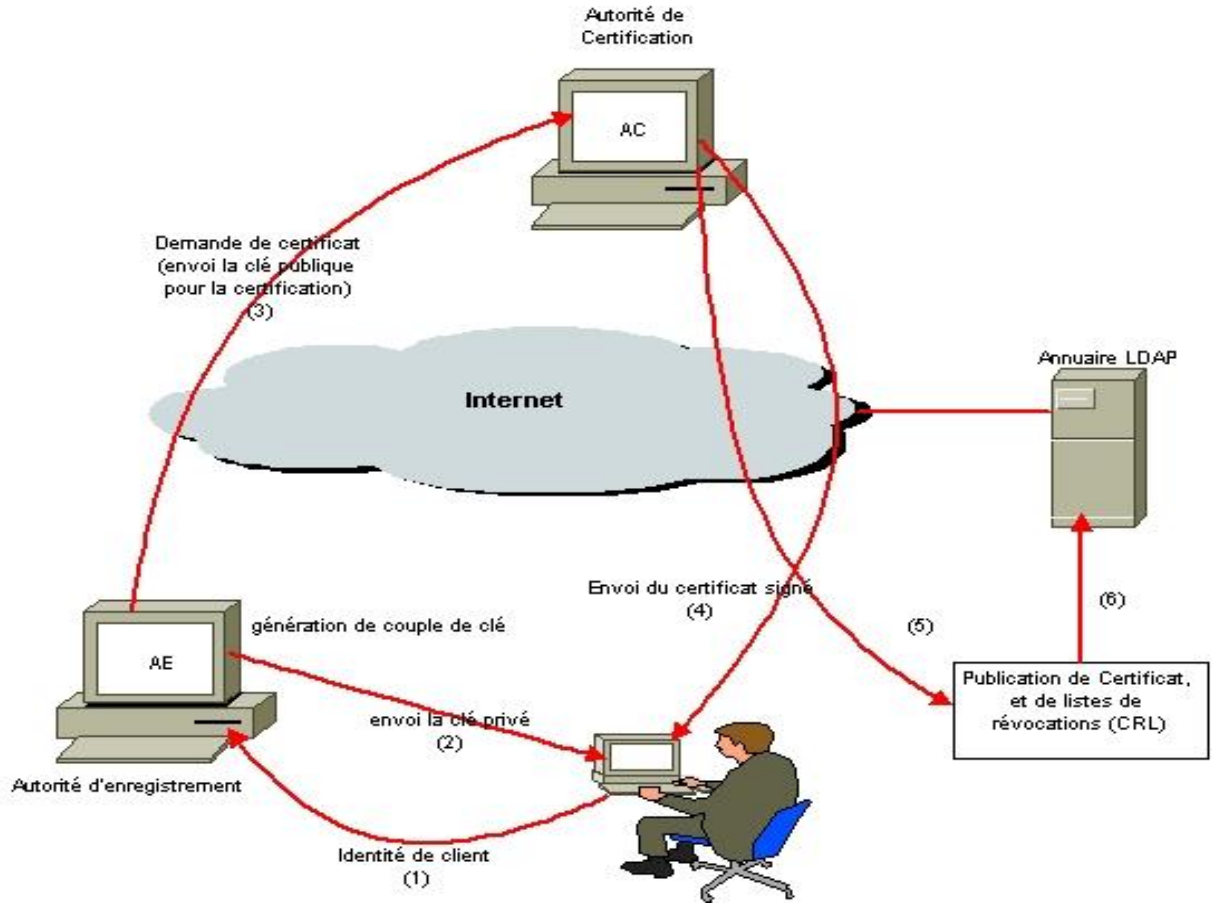


Figure 2: Organization d'une PKI

4.3.1. Les différents types de cartes

Il y a principalement deux grandes familles de carte :

- Les cartes à puce cryptographique qui nécessitent un lecteur. Elles permettent d'intégrer d'autres technologies pour d'autres usages, comme par exemple : une antenne sans contact (accès physique), ou une piste magnétique(cantine, badgeur).
- Les clefs USB (avec puce) qui n'ont pas besoin de lecteur et peuvent se connecter directement au PC avec les pilotes appropriés. Ces clefs USB peuvent apporter des fonctions complémentaires comme un disque externe.

4.3.2. L'infrastructure de PKI

Une infrastructure à clé publique est en règle générale composée de trois entités distinctes : L'**autorité d'enregistrement (AE)**. Cette entité est chargée des opérations administratives telle que la vérification de l'identité de l'utilisateur ou le suivi des demandes.

L'**autorité de certification (AC)**. Cette entité est chargée des tâches de création de certificats ou de signature des listes de révocation

L'**Autorité de dépôt (AD)**. Cette entité est chargée de la conservation en sécurité des certificats à des fins de recouvrement.

4.3.3. Les fonctions du CMS

Un Card Management System doit pouvoir effectuer les fonctions suivantes :

- Création d'une carte pour un nouvel employé : association de la carte à un employé et, dialogue avec l'AC de la PKI pour récupérer le certificat de l'employé et le mettre dans la carte
- Prêt d'une carte temporaire à un employé lorsque l'employé a oublié sa carte
- Mise en liste noire (blacklist) d'une carte perdue (ou retrait de la liste noire si elle est retrouvée)
- Déblocage en local ou à distance d'un code pin qu'un utilisateur a « verrouillé »Il doit être utilisable par le help desk pour gérer les fonctions de déblocage d'un code pin et par les structures d'accueil des différents sites pour la création et l'affectation d'une carte ou pour le prêt d'une carte.

4.3.4. Le module d'authentification

Le module d'authentification du poste doit permettre d'authentifier l'utilisateur :

1. Il demande l'identifiant et le code PIN de sa carte à l'utilisateur

2. Il vérifie auprès du CMS que la carte n'est pas dans la « liste noire » des cartes

3. Il récupère le certificat public dans la carte, vérifie sa signature et vérifie qu'il n'est pas publié dans la « liste noire »

4. Il demande à la carte de signer un *challenge* et vérifie (ou fait vérifier par un serveur) que la signature correspond bien au certificat public

En cas de validation des éléments, ce module autorise l'accès au poste de travail sous l'identité requise.

Il doit également gérer d'autres évènements :

- La perte ou l'oubli du code PIN. Le module d'authentification doit pouvoir permettre à l'utilisateur qui est au bout du monde et qui ne peut pas appeler son help-desk de récupérer un mot de passe ou un code pin en répondant à quelques questions.
- La réinitialisation à distance du code PIN d'une carte par le help-desk.

- La sécurisation du poste de travail lors du retrait de la carte : fermeture de la session, ou verrouillage simple.

4.4. L'identifiant et le mot de passe sur une carte à puce

Le stockage de l'identifiant et du mot de passe sur une carte à puce permet de compléter la sécurisation du processus d'authentification. Le mot de passe peut ainsi être très complexe et changé régulièrement de manière automatique et aléatoire. Sans la carte, et sans son code PIN, il n'y a plus d'accès au mot de passe. Cette solution est généralement mise en œuvre pour le processus d'authentification initiale.

4.5. Les solutions biométriques

Les solutions biométriques utilisent des lecteurs biométriques pour contrôler les accès physiques.

Il y a relativement peu de fournisseurs de lecteurs biométriques. Certains constructeurs de portable proposent une option pour intégrer ce type de lecteur dans le corps du portable.

Les solutions de biométrie sont en général utilisées à l'intérieur de l'entreprise pour protéger l'accès aux applications les plus sensibles. Il n'y a pas actuellement de normes appliquées par les navigateurs du marché qui permettraient de contrôler les accès à partir de n'importe quel PC sur internet.

4.5.1. Les trois familles de solution de biométrie

Le stoc kage et la gestion des données biométriques se sont heurtés aux règlementations régissant la protection de l'individu.

Les solutions de biométrie permettent de mettre en œuvre trois types différents d'architectures.

4.5.2. Les solutions de biométrie avec serveur

Elles s'appuient sur les composants suivants :

- Un serveur central,
- Un module d'enrôlement des signatures biométriques,
- Un module d'authentification spécifique pour gérer l'authentification.

4.5.3. Les solutions locales

Ces solutions évitent le stockage centralisé des signatures biométriques en stockanttoute donnée sensible sur le poste de l'utilisateur.
Si cette solution est plus acceptable d'un point de vue légal dans de nombreux pays,elle pose le problème de la mobilité des utilisateurs dans l'entreprise.

4.5.4. Les solutions de biométrie avec carte à puce

Ces solutions évitent également l'utilisation d'un serveur central, tout en donnant à l'utilisateur la possibilité de se déplacer au sein de l'entreprise. En effet, ses signatures biométriques sont conservées sur sa carte à puce et le suivent sur tous les postes de travail.
Si cette solution est à la fois plus sécurisée et mieux acceptée dans de nombreux pays, elle nécessite l'utilisation d'un « Card Management System » pour le déploiement des cartes et de disposer de tous les périphériques nécessaires sur les différents postes de travail.

4.6. L'identification sans contact

Le RFID est une technologie qui aujourd'hui se déploie dans les projets d'Identification/Authentification Une puce RFID est encastrée dans un badge et porte un numéro d'identification. Ce numéro est ensuite associé à un utilisateur dans un système informatique. A la base c'est une technologie d'identification qui peut, en étant couplée à un mot de passe fourni par l'utilisateur par exemple, être utilisé dans des procédures d'authentification. Il existe 2 déclinaisons de cette technologie :

4.6.1. Le RFID passif ou HID, qui suppose que la carte ne possède pas d'alimentation propre. La carte est alimentée lors de la lecture par un champ électromagnétique généré par le lecteur. Ce système est communément utilisé pour le contrôle d'accès physique par badge ou le paiement au restaurant d'entreprise. La détection d'une carte HID se fait à quelque centimètre.

4.6.2. Le RFID actif s'appuie sur les protocoles de communication RFID mais associe à la carte une alimentation propre. Cette alimentation permet une détection de la carte à plus longue portée (par exemple dès l'entrée dans une salle ou un bureau).L'intérêt principal du RFID actif est de permettre un constat d'absence pour les postes de travail dans des zones accessibles au public.***[09]***

5. Avantages et inconvénients des différentes techniques d'authentification

Le tableau suivant recense les principaux avantages et inconvénients de chacune des techniques d'authentification décrites précédemment :

Techniques	**Avantages**	**Inconvénients**
Mot de passe statique	+ facile à mettre en œuvre + facile à utiliser	- vol du mot de passe en regardant par-dessus l'épaule - oubli du mot de passe - peu robuste (facilement devinable ou « craquable ») - partageable, et trop souvent partagé
Mot de passe statique stocké dans une carte magnétique activée par code **PIN**	+ robustesse du mot de passe (possibilité de choisir un mot de passe aléatoire et comprenant des caractères spéciaux) + pas de nécessité de mémoriser le mot de passe	- vol, perte ou oubli de la carte - carte partageable
Mot de passe dynamique généré par un outil logiciel	+ robustesse du mot de passe (mot de passe souvent aléatoire et comprenant des caractères spéciaux) + confort d'utilisation pour l'utilisateur (absence de mémorisation du mot de passe)	- peu de confort d'utilisation (nécessité d'utiliser un logiciel à chaque nouvelle connexion) - protection de l'utilisation du logiciel par une personne non autorisée
Mot de passe dynamique généré par un outil matériel	+ confort d'utilisation pour l'utilisateur (absence de mémorisation du mot de passe) + robustesse du mot de passe (usage unique)	- vol, perte ou oubli du générateur de mot de passe - le générateur peut se désynchroniser avec le serveur qui contrôle la vérification du mot de passe
Certificat X.509 dans le navigateur	+ multi-usage	- vol ou utilisation frauduleuse

de l'ordinateur	+ robustesse de la méthode d'authentification + confort d'utilisation pour l'utilisateur	de l'ordinateur et copie de la clé privée associée au certificat
Certificat X.509 dans un token USB	+ multi-usage + robustesse de la méthode d'authentification + attitude similaire à la possession de clés (maison, voiture)	- vol, perte ou oubli du token
Certificat X.509 dans une carte à puce	+ multi-usage + robustesse de la méthode d'authentification + attitude similaire à la possession d'une carte bancaire	- vol, perte ou oubli de la carte à puce
Biométrie et caractéristiques de référence dans une base de données en réseau	+ pas d'oubli ou de vol possible	- technologie encore immature - facilement falsifiable
Biométrie associée à une carte magnétique		- technologie encore immature - vol, perte ou oubli de la carte magnétique - coût élevé
Biométrie associée à un certificat X.509 dans un token USB	+ multi-usage + attitude similaire à la possession de clés (maison, voiture)	- technologie encore immature - vol, perte ou oubli du token USB - coût élevé
Biométrie associée a un certificat X.509 dans une carte à puce	+ multi-usage + attitude similaire à la possession d'une carte bancaire	- technologie encore immature - vol, perte ou oubli de la carte à puce - coût élevé

Tableau 1: Avantages et inconvénients des différentes techniques d'authentification

6. Types et méthodes d'authentification

6.1. Authentification simple

Une authentification simple est une procédure d'authentification qui ne requiert qu'un seul facteur d'authentification.

6.2. Authentification forte

Une authentification forte est une procédure d'identification qui requiert la concaténation d'au moins deux facteurs d'authentification.

On peut le faire des multiples facteurs d'authentification pour augmenter la sécurité et la solidité de notre authentification.

Exemples de système d'authentification à **2** facteurs :

- Carte à puce + code PIN :(éléments **que l'on possède ET que l'on sait).**
- Carte à puce + biométrie :(élément **que l'on possède ET que l'on est).**
- Biométrie + mot de passe :(élément **que l'on est ET que l'on sait)).**

Exemple de système d'authentification à **3** facteurs :

- Carte à puce + code PIN + biométrie (éléments que l'on possède ET que l'on sait ET que l'on est).

Figure 3: Principe de l'authentification forte

La multiplication du nombre de facteurs d'authentification augmente le niveau de sécurité général, mais pose les problèmes suivants :

- Le cycle de vie de chaque facteur doit être géré : réinitialisation des mots de passe et codes PIN, distribution des cartes à puce, …,
- Les coûts des périphériques (cartes à puce, lecteurs, capteurs biométriques) sont additionnés. De plus, la charge du help-desk va s'accroître pour gérer l'ensemble de ces méthodes (déblocage des mots de passe et codes PIN, distribution des cartes, formation des utilisateurs à la biométrie, …).***[01]***

7. Conclusion

Dans ce chapitre on a mis la lumière sur la sécurité informatique spécifiquement sur le concept de l'authentification et leur importance dans la sécurité des applications web leurs techniques et leurs types de méthodes dans, le chapitre suivant on va voir la bonne solution pour l'authentification unifié et les outils qui sont adapté à notre travail.

Chapitre II : Les techniques d’authentification

1. Introduction

Compte tenu de la multiplication des applications web dans les établissements, les utilisateurs sont amenés à s'authentifier de nombreuse fois auprès de chacune de ces applications, en multipliant les couples identifiant/mot de passe à retenir. Le déploiement des annuaires LDAP, outre leur apport fonctionnel pour la gestion des groupes, a permis de simplifier la situation en utilisant un référentiel d'authentification commun à la majorité des applications. Ainsi pour les applications utilisant ce référentiel d’authentification, l’utilisateur peut utiliser un mot de passe unique. La mise en place d'un système de Single Sign-On (SSO) doit permettre à l'utilisateur de saisir ce mot de passe une seule fois pour accéder à toutes les applications web de l’établissement, améliorant ainsi à la fois l'ergonomie d'accès aux applications et la sécurité du système d'information en limitant la circulation des mots de passe.
En préalable à la description des architectures d’authentification, il importe de différencier deux fonctions souvent confondues : l’authentification et l’autorisation. L’authentification consiste à déterminer l’identité de l’utilisateur, alors que la fonction d’autorisation, pour une opération donnée, détermine les privilèges de l’utilisateur en fonction de ses attributs. Si certains de ces attributs sont gérés par les applications concernées, d’autres (comme les fonctions ou l’appartenance à des groupes) ont leur place dans l’annuaire LDAP de l’établissement.

A l’heure où les relations entre les établissements se développent (campus numériques, accès à des ressources bibliothécaires, étudiants inscrits dans plusieurs établissements, groupes de travail transversaux. L’utilisation du SSO comme service commun d’authentification au sein de l’établissement peut permettre de développer la gestion des identités entre établissements.

Aussi dans cette partie on met la lumière sur une autre architecture qui est le CAS (Central Authentification Service) est une architecture pour implémenter un système d'authentification unique (SSO) en s'appuyant sur des systèmes d'authentification tiers comme LDAP.

2. SSO (Single Sign On)

2.1. Définition

L'objectif du Single Sign On, noté SSO, est de centraliser l'authentification afin de permettre à l'utilisateur d'accéder à toutes les ressources (machines, systèmes, réseaux) auxquelles il est autorisé d'accéder, en s'étant identifié une seule fois sur le réseau. A terme, le SSO propage l'information d'authentification aux différents services du réseau, voire aux autres réseaux, et évite ainsi à l'utilisateur de multiples identifications par mot de passe.***[18]***

2.2.Architectures classique d'un SSO web

L'architecture de la plupart des produits de SSO est inspirée de Kerberos [2] ils utilisent largement sa terminologie et partagent ses concepts de base qui sont les suivants :

- Les applications sont déchargées du travail d'authentification des utilisateurs. Cette tâche est assurée par un serveur d'authentification dédié.
- Le serveur d'authentification délivre des tickets au client (maintien de la session d'authentification) et aux applications (transmission de l'identité de l'utilisateur). Ce second ticket transite également par le client.
- L'application ne recueille jamais les éléments d'authentification de l'utilisateur (couple identifiant + mot de passe par exemple).
- Il existe une relation de confiance entre les applications et le serveur d'authentification. A noter que Kerberos n'utilise que des techniques de cryptographie symétriques ; l'utilisation de certificats X509 (utilisant des algorithmes asymétriques) peut permettre de simplifier l'architecture du système.

Les principes de base d'un système de Single Sign-On web ont été par ailleurs définis dans des documents de référence. Compte tenu du développement par plusieurs universités américaines de systèmes de SSO « maison », Internet 2 a initié le groupe de travail WebISO chargé de définir les caractéristiques d'un système de Single Sign-On web. Le projet Liberty Alliance, quant à lui, est un regroupement d'entreprise (AOL, Cisco, France Telecom, HP, Novell, Sun, bVerisign,…) qui, en réaction face au projet « Microsoft Passport », publie des recommandations autour de la notion de « Federated Network Identity», une

[2]**Kerberos** est un protocole d'authentification réseau qui repose sur un mécanisme de clés secrètes (chiffrement symétrique) et l'utilisation de tickets, et non de mots de passe en clair.3

authentification unique avec partage des données de l'utilisateur avec des partenaires de confiance.

2.3. Le serveur d'authentification

Le serveur d'authentification est l'élément central du système de SSO puisqu'il assure l'authentification de l'utilisateur, la persistance de sa connexion et la propagation de l'identité de l'utilisateur auprès des applications.

L'utilisateur fournit ses éléments d'authentification au serveur d'authentification. Si le mode d'authentification est le mot de passe, la phase d'authentification implique la vérification du mot passe de l'utilisateur auprès d'une base de référence. La plupart des systèmes de SSO implémentent plusieurs backend d'authentification (/etc/password, NIS[3], LDAP). Si le serveur implémente l'authentification par certificats sa tâche consistera à vérifier la validité du certificat, la chaîne de certification et les listes de révocation. En concentrant la logique d'authentification sur un serveur d'authentification, on peut plus facilement faire évoluer les méthodes d'authentification (certificats, OTP,…) ainsi que les bases d'authentification reconnues (PAM, Radius,…).

Lorsque l'utilisateur a été authentifié, le serveur d'authentification maintiendra la session de l'utilisateur en positionnant un cookie HTTP[4] sur le poste de l'utilisateur. Les données stockées dans le cookie sont protégées (cryptage ou utilisation d'un ticket interprété par le serveur) et sa portée est idéalement limitée au serveur d'authentification. Le cookie HTTP est le seul moyen technique fiable à notre disposition pour que l'utilisateur soit reconnu comme authentifié lors de son prochain accès au serveur.

Si l'utilisateur a été orienté vers le serveur d'authentification par une application cible, le serveur doit, en retour, fournir l'identité de l'utilisateur à l'application. Par identité on entend soit uniquement un identifiant, un email et/ou un ensemble d'attributs de l'utilisateur. La transmission de l'identité de l'utilisateur à l'application transitera forcément par le poste de l'utilisateur. Soit le serveur d'authentification utilise une redirection HTTP, soit il renvoie à l'utilisateur un document HTML incluant un programme Javascript de redirection. Dans tous les cas le navigateur de l'utilisateur est redirigé vers l'application, muni des éléments d'identification. L'application ne fait appel qu'une fois au serveur d'authentification et gère ensuite une session applicative classique avec l'utilisateur.

[3]**NIS** : ***Network Information Service*** (**NIS**) est un protocole client serveur permettant la centralisation d'informations sur un réseau UNIX.

[4]**HTTP :** serveur HTTP ou daemon HTTP ou HTTPd ou serveur Web, est un logiciel servant des requêtes respectant le protocole de communication client-serveur Hypertext Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

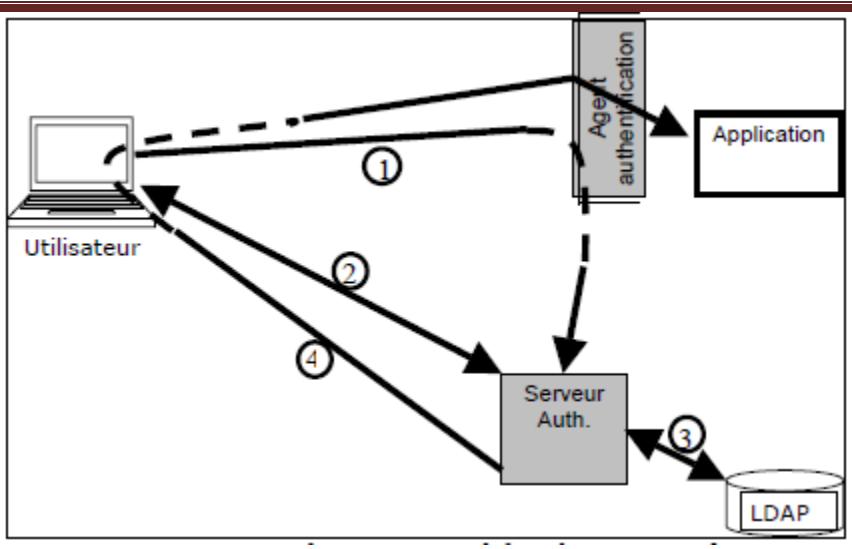


Figure 4: Architecture simple d'un SSO web

3. CAS(*Central Authentification Service*)

3.1. Définition

Développé par l'Université de Yale, CAS (*Central Authentication Service*) met en œuvre un serveur d'authentification accessible, CAS est une architecture pour implémenter un système d'authentification unique (SSO) en s'appuyant sur des systèmes d'authentification tiers comme LDAP, Active Directory, une base SQL, etc. L'architecture est générique et ne dépend pas du système d'authentification choisi. ***[19]***

3.2. Le mécanisme d'Architecture CAS

Le serveur CAS

L'authentification est centralisée sur une machine unique, le serveur CAS. Ce serveur est le seul acteur du mécanisme CAS à avoir connaissance des mots de passe des utilisateurs. Son rôle est double :

- **authentifier** les utilisateurs ;
- **transmettre et certifier l'identité** de la personne authentifiée (aux clients CAS).

*Les navigateurs (*web*)*

Les navigateurs doivent satisfaire les contraintes suivantes pour bénéficier de tout le confort de CAS :

- disposer d'un moteur de chiffrement leur permettant d'utiliser le protocole HTTPS.

- savoir effectuer des redirections HTTP(accéder à une page donnée dans un entête Location lors d'une réponse 30x à une première requête HTTP) et interpréter le langage JavaScript.
- savoir stocker des cookies. En particulier, les *cookies* privés ne devront être retransmis qu'au serveur les ayant émis pour garantir la sécurité du mécanisme CAS.

Ces exigences sont satisfaites par tous les navigateurs classiquement utilisés, à savoir *Microsoft Internet Explorer* (depuis 5.0), *Netscape Navigator* (depuis 4.7) et *Mozilla*.

3.3. Les clients CAS

Une application web muni d'une librairie cliente ou un serveur web utilisant le module *mod_cas* est alors appelé « client CAS ». Il ne délivre les ressources qu'après s'être assuré que le navigateur qui l'accède se soit authentifié auprès du serveur CAS. Parmi les clients CAS, on trouve :

- des librairies correspondant aux langages communément employés en programmation web dynamique (*Perl*, *Java*, JSP,PHP, ASP) .
- un module *Apache*, qui permet de protéger des documents statiques.
- un module PAM, qui permet d'authentifier les utilisateurs au niveau système.

3.4. Authentification d'un utilisateur

Un utilisateur non déjà précédemment authentifié, ou dont l'authentification a expiré, et qui accède au serveur CAS se voit proposer un formulaire d'authentification, dans lequel il est invité à entrer son nom de connexion et son mot de passe :

Si les informations sont correctes, le serveur renvoie au navigateur un *cookie* appelé TGC (*Ticket Granting Cookie*)

Le *Ticket Granting Cookie* (TGC) : est le passeport de l'utilisateur auprès du serveur CAS. Le TGC, à durée de vie limitée (typiquement quelques heures), est le moyen pour les navigateurs d'obtenir auprès du serveur CAS des tickets pour les clients CAS sans avoir à se ré-authentifier. C'est un *cookie* privé (n'est jamais transmis à d'autres serveurs que le serveur CAS) et protégé (toutes les requêtes des navigateurs vers le serveur CAS se font sous HTTPS). Comme tous les tickets utilisés dans le mécanisme CAS, il est opaque (ne contient aucune information sur l'utilisateur authentifié) : c'est un identifiant de session entre le navigateur et le serveur CAS.***[10]***

4. L'authentification unifiée

Le concept d'authentification unifiée c'est le concept qu'on a utilisé dans notre mémoire se compose d'un annuaire LDAP qui est un référence fournissent un référentiel d'informations sur les autre plateformes. Le référentiel central utilisé pour les données LDAP qui est connecté et bien configurer avec le serveur de messagerie Zimbra et les plateformes Dokeos et Joomla , cette authentification va se concentrer beaucoup sur le travail du serveur LDAP qui regroupe tous les comptes des utilisateurs dans un seul annuaire , avec ce dernier l'utilisateur doit s'authentifier et accéder au plateformes , donc c'est une authentification unifiée qui unifier ces plateformes Zimbra , Joomla (CMS), Dokeos (LMS) .

5. Conclusion

Dans ce chapitre on a parlé de SSO single Sign-ON son objectif et ses différentes architectures et le CAS qui est une autre architecture de l'authentification unifiée et pour le chapitre suivant on détermine les outils nécessaires pour notre implémentation.

Chapitre III :Les outils d'authentification unifiée

Les outils utilisés dans notre travail

1. Introduction :

Comme on a parlé dans le chapitre précédent sur les techniques d'authentification qui est le SSO (singl-sign on) et le CAS (central authentification service) dans ce chapitre on commence à présenter les outils utiles à l'authentification, pour notre travail on a montré les outils utilisable à notre environnement universitaire qui est premièrement le serveur LDAP c'est l'annuaire pour la base de données des utilisateurs avec le messagerie électronique Zimbra et les deux plateformes Dokeos et Joomla .

2. Présentation de l'environnement

2.1. Le CentOS :

CentOS est une distribution dérivée de RedHat Entreprise Linux. Son avantage est de proposer la stabilité de RedHat sans avoir à souscrire un contrat de support, il était au final entièrement une open-source parmi ces nouveautés on notera l'amélioration des outils de virtualisation ou les outils de développements et de monitoring qui intéresseront particulièrement les développeurs C++ et Python. Dans ce domaine Cent OS 6.1 propose

également une version mise à jour d'Eclipse.
Autre point mis en avant par le site officiel, Modified (Yum) le gestionnaire de paquets en ligne de commande. Bien que restant très proche de RHEL 6.2 (Red Hat Enterprise Linux), la communauté de CentOS a également modifié le Kernel Linux, ainsi que les paquets Firefox et Apache HTTP.

2.1.1. Pourquoi choisir la distribution CentOS Enterprise Linux ?

La question qu'un utilisateur qui n'a jamais installé le système CentOS Linux, distribution Serveur, peut poser est : Pourquoi choisir CentOS Linux plutôt qu'une autre distribution Linux Serveur ? Avant de répondre à cette question, permettons –nous de révéler un peu l'historique de ce système d'exploitation.
Le toute premier release du système CentOS (Community Enterprise Operating System) créé par le groupe CentOS Développent Team est sortie au mois de mai 2004. Etant depuis une distribution 100% Open Source et totalement gratuite, CentOS est basée sur la distribution

RedHat Entreprise Linux (RHEL). Elle utilise les sources de la RHEL (téléchargeables librement sur Internet) pour régénérer la RedHat à l'identique. On peut donc considérer la CentOS comme une version gratuite de la RedHat. Le support technique est alors de type communautaire : il se fait gratuitement et ouvertement via les listes de diffusion et les forums de la communauté CentOS.

Revenons maintenant à la fameuse question: Pourquoi choisir ce système ?

Chaque système d'exploitation a ses qualités et ses défauts. Il faut tout simplement les distinguer et faire ensuite son propre choix par rapport à ses besoins et attentes. Voici donc les avantages qui m'ont fait porter notre choix sur cette distribution :

- Support gratuit. Mises à jour applicatives et les patchs de sécurité réguliers.
- Stabilité quasi-équivalente à la distribution RedHat utilisé dans de gros environnements de production.
- Cycle de développement suivant celui de RedHat (7ans pour un release).
- L'outil "YUM" facilitant l'exploitation et la gestion des paquets au format RPM.
- Nombreux manuels en ligne (en anglais et en français) de RedHat, 100% compatibles CentOS Linux.

Bien sûr, il n'y a pas d'avantages sans inconvénients, mais il faut vivre avec :

- Limite au niveau des dépôts standards fournissant les paquets RPM.
- Difficulté de création de ses propres paquets RPM.[11]

2.2. Annuaire LDAP

2.2.1. Introduction

L'informatique et la gestion de l'information prend une place de plus en plus importante dans notre société, particulièrement en entreprises. La multiplication des applications et des serveurs rend cette information difficile à maîtriser car très volatile et éparse. Ceci entraîne bien souvent une obsolescence, voire une incohérence des données stockées. Les annuaires LDAP offrent une réponse à ce problème en proposant de centraliser les informations.

2.2.2. Historique et aperçu des annuaires existants

En 1988, l'Union Internationale des Communications (UIT) met au point les annuaires X.500. Le but de cette opération est d'uniformiser l'accès aux services, de centraliser les ressources et de les protéger. Le protocole utilisé pour y accéder est le protocole DAP (Directory AccessProtocol).

Malheureusement, le protocole DAP s'avère difficile à mettre en œuvre et ne fonctionne pas sur les réseaux TCP/IP. En 1993, l'Université du Michigan réfléchit donc à un moyen de pallier ces deux problèmes : elle met en place le protocole LDAP (Lightweight Directory AccessProtocol), au départ simple "connecteur" TCP/IP avec des annuaires X.500.

En 1995, LDAP devient un protocole natif et utilisable indépendamment de X.500.

LDAP est donc une évolution de la norme X.500. Sa version actuelle est la version 3 (RFCs 2251, 4511, 4512, 4513), elle propose les évolutions suivantes par rapport à la version 2 :

• Le support des communications chiffrées via SSL/TLS
• L'authentification via SASL
• Le support des Referrals (une branche pointe vers un autre annuaire)
• Le support d'Unicode (internationalisation)
• La capacité d'étendre le protocole
• Le support des schémas dans l'annuaire

2.2.3.Types d'annuaires

D'autres types d'annuaires existent, vous les utilisez très certainement :

• DNS : Domain Name Services
• NIS : Network Information Services
• Whois : base d'information concernant les noms de domaines

2.2.4.Les annuaires LDAP

Voici une liste des principaux annuaires LDAP existant sur le marché :

• Open LDAP
• Apache Directory Server
• Sun (One/Java) Directory Server
• Active Directory

2.2.5. Les concepts du protocole LDAP

On a coutume de regrouper les caractéristiques et fonctionnalités de l'annuaire LDAP sous la forme de quatre modèles :

• Le modèle de nommage : définit comment l'information est stockée et organisée
• Le modèle fonctionnel : définit les services fournis par l'annuaire (recherche, ajout, ...)
• Le modèle d'information : définit le type d'informations stockées
• Le modèle de sécurité : définit les droits d'accès aux ressources

2.2.6. Le protocole

LDAP signifie "Lightweight Directory Access Protocol".
LDAP est un protocole, ce qu'il signifie que son rôle est de présenter des informations. Un serveur LDAP agit en tant qu'intermédiaire entre une source de données et un client.
Nous verrons qu'en tant qu'intermédiaire il définit quelques conventions, notamment l'organisation des données qu'il présente qui sera sous forme hiérarchique, mais aussi un format d'échange standard.
LDAP fonctionne sur le port TCP 389 (par défaut).

2.2.7. Organisation des données (modèle de nommage)

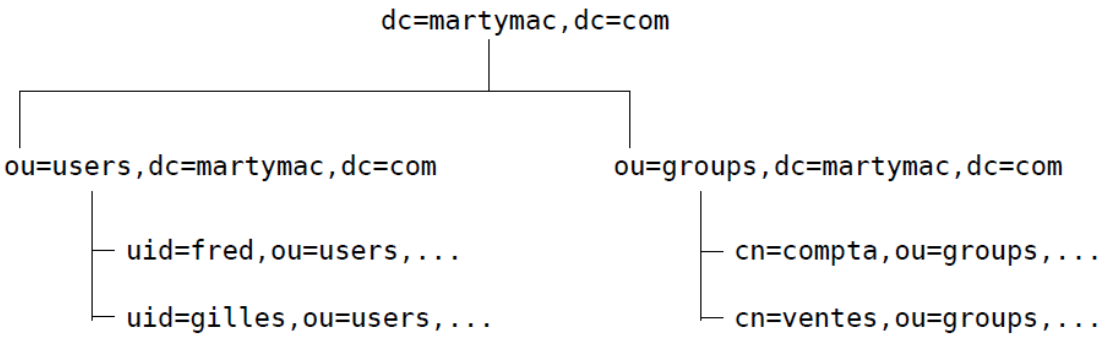
2.2.7.1. Introduction

Le modèle de nommage est la manière dont sont organisées les données dans l'annuaire. Etudions cette organisation plus en détails...

2.2.7.2. La représentation hiérarchique des données

LDAP organise les données de manière hiérarchique dans l'annuaire. Ceci signifie que toutes les informations découlent d'une seule et même "racine".

Voici un exemple d'arborescence LDAP :



Cette arborescence est liée au nommage de chaque élément : un élément marque son appartenance à l'élément supérieur en reprenant le nom, qu'il complète par le sien.

Ainsi, en étudiant simplement le nom de l'élément :

"cn=ventes, ou=groups, dc=martymac, dc=com."

il est possible de le situer dans la hiérarchie : il est situé sous l'élément "**ou=groups**" qui lui-même est situé sous l'élément "**dc=martymac, dc=com.**".

2.2.7.3. Termes à connaître

Voici quelques termes :

• Chaque élément est appelé une **entrée** (an entry). Une entrée peut être un branchement (un **noeud**) ou un élément terminal (une **feuille**).

• Chaque élément possède un **DN** (Distinguished Name). Le DN est le nom complet de l'élément qui permet de le positionner dans l'arborescence. Il est unique dans l'annuaire.

Exemple : "cn=ventes, ou=groups, dc=martymac, dc=com"

• Chaque élément possède également un **RDN** (Relative Distinguished Name). Le RDN est la partie du **DN** de l'élément qui est relative au **DN** supérieur. Le RDN d'un élément ne permet pas de l'identifier de manière absolue dans l'annuaire.

Exemple : "cn=ventes"

• La **racine** est l'élément supérieur de tous les autres, c'est la base de l'arborescence. On l'appelle **root** en anglais, parfois on parle de "**root DN**".

Exemple : "dc=martymac, dc=com"

Les DN de chaque entrée sont composés au moins d'un attribut de l'élément (par exemple

"cn" ou "uid") et de sa valeur. Un attribut est l'une des caractéristiques de cet élément.
Remarquez que la racine choisie ici est composée du nom du domaine où est hébergé notre serveur LDAP, martymac.com, décomposé en "dc" (Domain Components) pour obtenir dc=martymac, dc=com .
L'arbre se découpe ensuite en deux "ou" (Organisational Units) qui constituent deux branchements : "users" et "groups", dans lesquels nous trouvons ensuite les entrées feuilles de notre arbre : les utilisateurs et les groupes.
Chacune des entrées de notre arbre correspond à un type de donnée particulier, défini par une classe d'objet. Nous étudierons ces notions par la suite.

2.2.7.4. Règles de nommage

La RFC 2253 (rendue obsolète par la RFC 4514) normalise l'écriture des DN et conseille de nepas ajouter d'espaces autour du signe "=", ni à la fin du DN. Les espaces sont autorisés par contre pour les valeurs des entrées. Ainsi, le DN suivant est correct :
"cn=Ganael Laplanche, cn=ventes, ou=groups, dc=martymac, dc=com"
Alors que celui-ci ne l'est pas :
"cn = Ganael Laplanche, cn = ventes, ou = groups, dc = martymac, dc = com"
Les majuscules seront ou non prises en compte en fonction du type d'attribut utilisé et de ses particularités.

2.2.8. Accéder à l'annuaire (modèle fonctionnel)

Il existe plusieurs types d'opérations que l'on peut effectuer sur l'annuaire, voici les plus importantes :
• Rechercher une entrée suivant certains critères
• S'authentifier
• Ajouter une entrée
• Supprimer une entrée
• Modifier une entrée
• Renommer une entrée
Certaines de ces actions, notamment la recherche, nécessitent des outils particuliers pour nous faciliter l'accès à l'annuaire

2.2.8.1. La base

La base est le DN à partir duquel nous allons agir. Pour une recherche, il s'agir du nœud à partir duquel est effectuée la recherche. Il peut s'agir de la racine de l'arbre pour une recherche sur la totalité de l'arbre, par exemple "dc=martymac, dc=com".

2.2.8.2. La portée

La portée (scope) est le nombre de niveaux sur lesquels l'action va être effectuée. Il existe 3 niveaux différents :

• SUB : l'action est effectuée récursivement à partir de la base spécifiée sur la totalité de l'arborescence.

• ONE : l'action est effectuée sur un seul niveau inférieur par rapport à la base spécifiée (les fils directs). Si l'on effectuait une recherche avec la portée ONE à partir de "dc=martymac, dc=com", nous pourrions trouver "ou=users, dc=martymac, dc=com" et "ou=groups, dc=martymac, dc=com".

• BASE : l'action est effectuée uniquement sur la base spécifiée. Une recherche sur "dc=martymac ,dc=com" avec la portée BASE renverrait cette entrée uniquement.

2.2.8.3. Les filtres

Le troisième outil à notre disposition est le filtre. Un filtre va permettre d'effectuer des tests de correspondance lors d'une recherche. Il s'agit en quelques sortes du critère de la recherche.
Il existe 4 tests basiques, qui peuvent ensuite être combinés :

• Le test d'égalité :**X=Y**

• Le test d'infériorité :**X<=Y**

• Le test de supériorité :**X>=Y**

• Le test d'approximation : **X~=Y**

Les autres opérateurs (<, >) ou des tests plus complexes peuvent être mis en place par combinaison, il faut alors utiliser les parenthèses ( ) et l'un des opérateurs suivants :

• L'intersection (et) : **&**

• L'union (ou) : **|**

• La négation (non) : **!**

Un test d'infériorité stricte pourrait donner ceci : (&(X<=Y)(!(X=Y)))
On peut combiner plus de deux éléments : (&(X=Y)(Y=Z)(A=B)(B=C)(!(C=D)))

Ces filtres seront appliqués sur des attributs choisis pour sélectionner finement les données que nous voulons extraire de notre annuaire.

2.2.8.4. Les URLs LDAP

Récemment est apparue une méthode concise et simplifiée pour interroger un annuaire LDAP. Il s'agit d'un format d'URL combinant toutes les notions que nous avons étudiées. En une seule ligne, il est possible de spécifier tous les éléments de notre requête. Voici le format de cette URL (RFC 2255, rendue obsolète par la RFC 4516) :

ldap[s]://serveur[:port][/[base[?[attributs à afficher][?[portée][?[filtre][?extensions]]]]]]

L'exemple ci-dessous recherche tous les uid de notre arbre, à partir de la branche users :

ldap://localhost:389/ou=users,dc=martymac,dc=com?uid?sub

2.2.9. Les données contenues dans l'annuaire (modèle d'information)

2.2.9.1. Les attributs

Nous avons jusqu'ici évoqué la notion d'attribut sans trop l'expliquer. Un attribut est une valeur contenue dans une entrée. Une entrée peut bien entendu contenir plusieurs attributs. Prenons l'exemple de l'entrée LDAP complète d'un compte utilisateur POSIX :

```
dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
```

Figure 5: Entrée complète avec le format LDIF

Ceci correspond à une entrée complète, extraite par une interrogation de l'annuaire. Le format affiché est le format **LDIF**.

Ce paragraphe présente tous les attributs, un par ligne, que comprend notre entrée. Un attribut est séparé de sa valeur par ":". Suivant son type, un attribut peut avoir plusieurs valeurs : dans ce cas, il est dit "multi-valué" et apparaît sur plusieurs lignes avec des valeurs différentes.

Nous pouvons observer ici des attributs nommés "dn", "objectClass", "cn", "uid", ...

L'attribut "dn" qui est indiqué en première ligne est le nom unique de notre entrée dans l'arbre dont nous avons parlé précédemment. Il constitue un attribut à part entière dans notre entrée.
Il est composé du dn de l'entrée supérieure, ainsi que du rdn.

Sur un annuaire LDAP la racine est toujours composée des attributs "dc" (Domain Component)associés à chacune des parties du nom de domaine où est hébergé le serveur ("dc=martymac, dc=com" pour le domaine martymac.com). Ceci est une convention. X500 préconisait les attributs "o", "l" et "c", mais LDAP a simplifié le procédé (cf. RFCs 2247, 4519,4524).L'attribut "ou" constitue une "Organisational Unit", c'est à dire une unité organisationnelle : en quelque sorte un regroupement. Nous avons choisi d'en créer deux dans notre exemple :
"users", qui accueillera nos utilisateurs et "groups", nos groupes.
Nous n'allons pas étudier chacun des attributs présents ici, cependant, nous souhaiterons porter votre attention sur l'un des attributs les plus importants, il s'agit de la classe d'objet, ou "objectClass"...

2.2.9.2. Les classes d'objets

A première vue, l'entrée présentée ci-dessus constitue un amalgame de différentes informations qui ne semblent pas organisées. Toutes ces entrées sont induites par la présence des objectClass.
L'objectClass d'une entrée est un attribut qui permet de cataloguer cette entrée. Un objectClass définit un regroupement d'attributs obligatoires ou autorisés pour une entrée.
Une entrée peut posséder un ou plusieurs objectClass. Ce sont ces objectClass qui définissent la présence de tous les autres attributs.
Ici, l'objectClass "posixAccount" rend obligatoire les attributs cn, uid, uidNumber, gidNumber et homeDirectory. Il rend possible l'utilisation des 4 autres attributs userPassword, loginShell, gecos et description.

2.2.9.3. Les schémas

Comment savoir quels sont les objectClass disponibles et quels attributs ils contiennent. C'est très simple, la syntaxe et la liste des attributs connus de l'annuaire sont écrits dans ce que l'on appelle les "schémas". Un annuaire LDAP a la capacité de charger en mémoire plusieurs schémas. A travers ces schémas, il est possible de définir de nouveaux attributs et de

nouveaux objectClass. Cette souplesse permet de définir très finement ce qui sera stocké dans notre annuaire.

Concrètement, un schéma est un fichier qui décrit un à un les attributs disponibles (leur nom, leur type, etc...), ainsi que les objectClass qui y font appel. Au démarrage du serveur LDAP, le ou les fichiers de schéma spécifiés dans sa configuration seront chargés.

Dans notre exemple, l'objectClass posixAccount est défini dans le fichier **nis.schema**. Etudions une partie de ce fichier, livré avec OpenLDAP et situé dans **/etc/ldap/schema** :

```
# [...]
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
        DESC 'An integer uniquely identifying a user in a domain'
        EQUALITY integerMatch
        SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )

# [...]
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
        DESC 'Abstraction of an account with POSIX attributes'
        MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
        MAY ( userPassword $ loginShell $ gecos $ description ) )
```

Figure 6: Attribut utilise par posixAccount et le objectClass

Le fichier est assez volumineux et a été tronqué.

Le premier paragraphe définit l'un des attributs utilisés par le posixAccount : uidNumber. Le second, l'objectClass posixAccount. Nous n'allons pas étudier en détail ces deux définitions, simplement, sachez que :

• A chaque définition correspond un OID (Object IDentifier), qui permet de rendre unique l'attribut spécifié. Ces OIDs sont déposés auprès de l'IANA et sont donc officiels.

• Un attribut définit un type d'égalité à mettre en œuvre lors d'une recherche (ici, intergerMatch) ainsi que le type de données qu'il contient (l'OID spécifié après SYNTAX).

• Un objectClass définit les attributs que l'objet doit présenter (MUST) et ceux qu'il peut posséder (MAY).

Les schémas constituent donc une source d'information très importante. En cas de doute concernant le type ou le nom des attributs à spécifier dans une entrée, n'hésitez pas à vous y reporter. Enfin, sachez qu'il est tout à fait possible de créer ses propres schémas, cependant, il faut penser à réutiliser les schémas existants .

2.2.9.4. Le format LDIF

Les données contenues dans l'annuaire sont présentées dans un certain format : il s'agit du format LDIF (LDAP Data Interchange Format - RFC 2849). Nous en avons vu un exemple dans le paragraphe précédent.

Sachez que toute interaction avec un annuaire se fait par le biais de ce format : l'ajout, la modification, la suppression d'entrées, l'interrogation de l'annuaire y compris.

Dans ce format, chaque entrée constitue un paragraphe, et, au sein de chaque paragraphe, chaque ligne constitue un attribut. Voici un exemple un peu plus complet, incluant le groupe de notre utilisateur :[04]

```
# [...]
dn: cn=utilisateurs,ou=groups,dc=martymac,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 10001

dn: uid=martymac,ou=users,dc=martymac,dc=com
objectClass: account
objectClass: posixAccount
cn: martymac
uid: martymac
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/martymac
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: martymac
description: martymac
# [...]
```

Figure 7: Entrée incluant le groupe avec le format LDIF

2.3. Plateforme

Une plate-forme est en informatique une base de travail à partir de laquelle on peut écrire, lire, développer et utiliser un ensemble de logiciels.

Les plates-formes informatiques sont généralement conçues, développées, construites, mises en service et maintenues par des constructeurs informatiques, ou des prestataires de services. Dans le cas des plates-formes logicielles, elles sont plutôt maintenues par les organismes (par exemple l'INRIA, le CNRS, le CEA, l'INRA) qui hébergent la base de travail et les logiciels associés.

Lorsqu'on parle de plate-forme web, il peut s'agir du logiciel serveur web, de ce même logiciel avec son système d'exploitation sous-jacent, du logiciel serveur web avec son système d'exploitation et son matériel, d'un ensemble de machines avec serveur web, ou encore d'un même ensemble en tenant compte des infrastructures réseau et connectivité à Internet.

2.3.1. Zimbra

2.3.1.1. Introduction

Le courrier électronique est l'une des applications les plus indispensables à la viabilité d'une petite ou moyenne entreprise. Outil de communication à son origine, il est devenu la solution d'archivage de fait pour les données métier dans de nombreuses entreprises, et sert de plate-forme quasi universelle de messagerie, coordination et collaboration. Son évolution va se poursuivre :

• *Fonctionnalités* : les attentes des utilisateurs en termes d'expérience homogène et facile s'avèrent élevées, et les préférences d'interface utilisateur fortes

• *Plates-formes* : à l'origine installé sur des serveurs d'entreprise, le courrier électronique fut l'une des premières applications à adopter la virtualisation , les périphériques mobiles et le Cloud

Les attentes croissantes des utilisateurs se heurtent constamment aux barrières économiques du coût de la solution et des tâches d'administration. Alors que les contrats de support arrivent à expiration pour certaines solutions de messagerie répandues, de nombreuses entreprises recherchent des alternatives.

2.3.1.2. Définition de Zimbra

Zimbra est un logiciel serveur collaboratif (ou groupware) qui permet à ses utilisateurs de stocker, organiser et partager rendez-vous, contacts, courriels, liens, documents et plus.

Zimbra offre une solution éprouvée dans des environnements de production, un choix d'options de déploiement en local avec gestion intégrée du stockage hiérarchique ou un hébergement par l'un des nombreux partenaires fournisseurs de services VMware vCloud. Zimbra propose également une appliance virtuelle logicielle reposant sur VMware vSphere, qui se déploie en moins de 10 minutes, dotée d'une interface d'administration simplifiée et conjuguant l'application et le système d'exploitation dans une seule procédure de gestion du

cycle de vie pour réduire les tâches de maintenance. L'appliance virtuelle de collaboration Zimbra utilise la plate-forme vSphere pour assurer une haute disponibilité, une sauvegarde et une reprise d'activité intégrées dans une véritable solution métier.
La messagerie Zimbra est conçue pour fonctionner de manière optimale sur les principales plates-formes informatiques (matériel, OS, virtualisation et Cloud), et avec des applications intégrées et des services Web hébergés . Sécurité, extensibilité, évolutivité et pérennité : Zimbra est une solution ouverte et simple à gérer à laquelle font confiance des millions d'utilisateurs, d'entreprises et de prestataires de services dans le monde entier. S'adossant aux ressources et l'assistance de VMware et ses partenaires, Zimbra incarne un choix sûr.

VMware Zimbra est un leader des logiciels de messagerie et de collaboration open source de nouvelle génération. Zimbra simplifie l'informatique et s'établit en référence de la collaboration sur le Web et le Cloud avec une expérience utilisateur novatrice et évolutive, intégrant une interface Web AJAX enrichie. Administration simplifiée, mobilité avancée et options de déploiement en local ou en hébergement sur le Cloud : Zimbra est une plate-forme de collaboration privilégiée pour les entreprises, les prestataires de services, les services publics et le monde de l'enseignement. Zimbra est l'un des principaux fournisseurs de messagerie, en croissance rapide.***[02]***

2.3.1.3. Zimbra Composants

L'architecture Zimbra inclut des intégrations open-source à l'aide standard de l'industrie protocoles. Le logiciel tiers énuméré ci-dessous est fourni avec Zimbra logiciels et installé dans le cadre du processus d'installation. Ces composant sont été testés et configurées pour fonctionner avec le logiciel.

- Jetty, le serveur d'applications Web que le logiciel Zimbra doit fonctionner
- Postfix, un agent de transfert de courrier open source (MTA) qui achemine le courrier messages vers le serveur Zimbra approprié
- logiciel Open LDAP, une implémentation open source du Lightweight

Directory Access Protocol (LDAP) qui stocke la configuration système Zimbra, la liste d'adresses globale Zimbra, et les fournisseurs d'authentification de l'utilisateur. Zimbra

peuvent également travailler avec les services de GAL et authentification fournies par externe Annuaires LDAP comme Active Directory

- la base de données du logiciel MYSQL
- Lucene, une open source texte complet et moteur de recherche
- Anti-virus et des composants open source anti-spam, contient:

• ClamAV, un scanner anti-virus qui protège contre les fichiers malveillants

• SpamAssassin, un filtre de messagerie qui tente d'identifier le spam

•James / Sieve de filtrage, utilisé pour créer des filtres pour le courrier électronique

2.3.1.4. Architecture du système

La conception architecturale ZCS est affichée dans la Figure de l'architecture ZCS serveur de collaboration. Cela montre le logiciel open-source livré avec le ZCS et d'autres applications tierces recommandées.

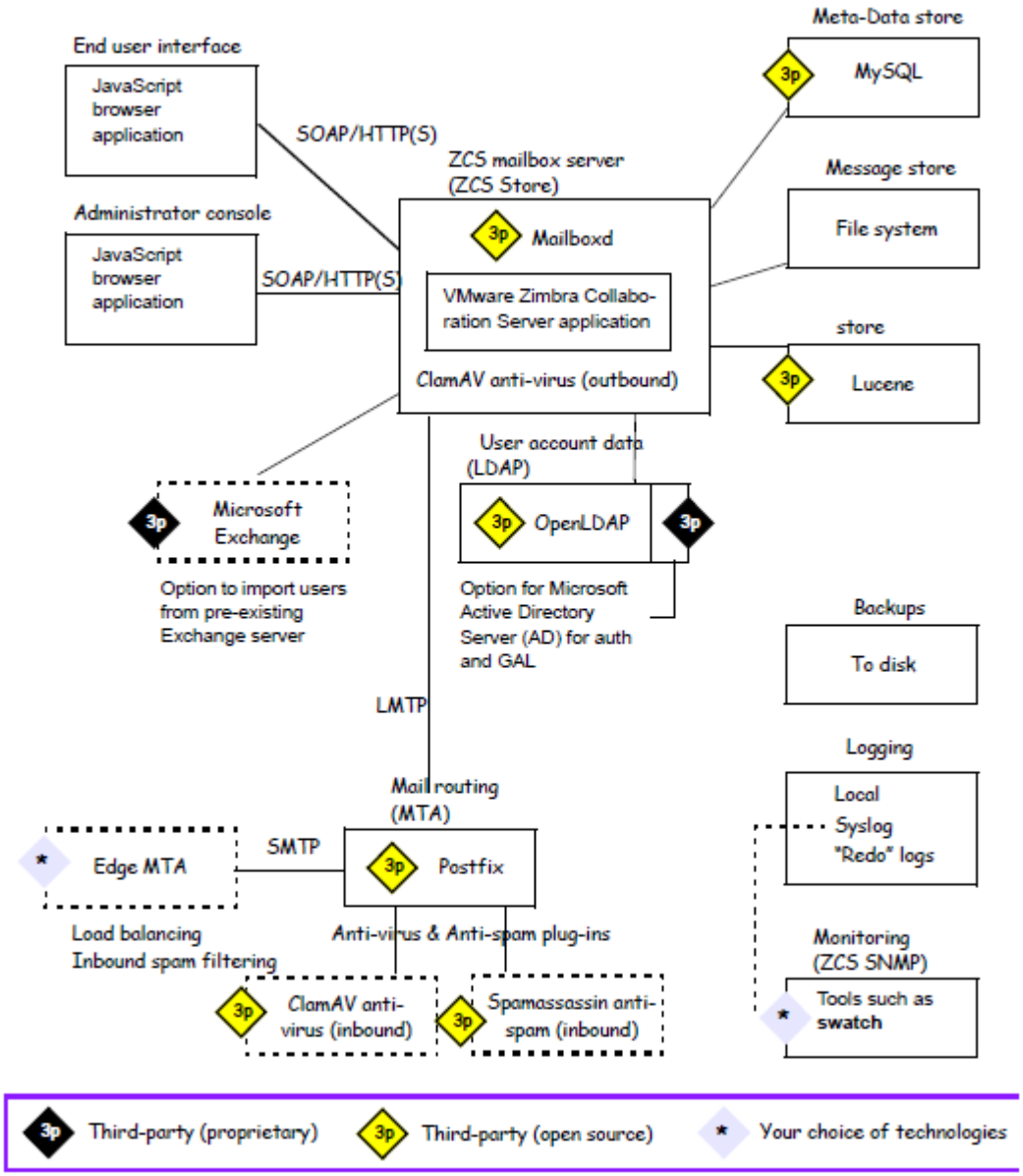


Figure 8: Architecture collaborative du serveur ZCS

2.3.1.5. Le packages d'applications Zimbra

Les paquets d'application inclut a ZCS :

2.3.1.5.1. Zimbra Core(base) : Comprend les bibliothèques, utilitaires, outils de suivi et les fichiers de configuration de base *.zmconfigd* fait partie de Zimbra-core et est automatiquement activée et fonctionne sur tous les systèmes.

2.3.1.5.2. Zimbra LDAP : ZCS utilise le logiciel OpenLDAP, une open source du serveur d'annuaire LDAP. L'authentification des utilisateurs, la liste d'adresses globale Zimbra et les attributs de configuration sont les services fournis par OpenLDAP. Notez que le GAL Zimbra et services d'authentification peuvent être fournis par un annuaire LDAP externe comme Active Directory.

2.3.1.5.3. Zimbra MTA : Postfix est l'agent de transfert de courrier open source (MTA) qui reçoit l'email via SMTP et routes chaque message au serveur de messagerie Zimbra approprié à l'aide locale Mail Transfer Protocol (LMTP).Le MTA Zimbra inclut également l'anti-virus et anti-spam composants.

2.3.1.6. Stockage de Zimbra (serveur de messagerie)

Le paquet du stockage Zimbra installe les composants du serveur de boîte aux lettres, y compris Jetty, qui est le conteneur de servlets le logiciel Zimbra gère l'intérieur. Dans ZCS, cette conteneur de servlet est appelée mailboxd. Chaque compte est configuré sur un serveur de boîte aux lettres, et ce compte est associé à une boîte aux lettres qui contient tous les messages électroniques, les pièces jointes, contacts, agenda et les fichiers de collaboration pour ce compte de messagerie.

Chaque serveur Zimbra a son propre stockage autonome de données, stockage de messages, et un stockage d'index pour les boîtes aux lettres sur ce serveur.

Comme chaque e-mail arrive, les horaires de serveur Zimbra un fil pour que le message soit indexé (Index stockage).

Zimbra Zimbra-SNMP : utilise échantillon de regarder la sortie de syslog pour générer des interruptions SNMP.

Zimbra-Logger : Le Zimbra logger installe des outils d'agrégation de syslog, rapports. Si l'enregistreur n'est pas installé, la section des statistiques du serveur de la console d'administration n'est pas affichée.

Zimbra-Spell Aspell : est le correcteur orthographique open source utilisé sur le Zimbra Web Client. Quand Zimbra-sort est installé, le Paquet Zimbra-Apache est également installé.

Zimbra-Proxy : L'utilisation d'un serveur proxy IMAP / POP permet la récupération de courrier pour un domaine à être divisée entre plusieurs serveurs Zimbra sur une base d'utilisateur.

Le paquet de Proxy Zimbra peut être installé avec le LDAP Zimbra , le Zimbra MTA, le serveur de messagerie Zimbra, ou sur son propre serveur.

Zimbra-Memcached est un paquet séparé de zimbra_proxy et est automatiquement sélectionné quand le paquet Zimbra-proxy est installé. Un serveur doit exécuter zimbra_memcached lorsque le proxy est en cours d'utilisation. Tous zimbraproxies installés peuvent utiliser un seul serveur de cache.

2.3.1.7. Service de LDAP Zimbra

Services d'annuaire LDAP fournissent un référentiel centralisé d'informations sur les utilisateurs et les périphériques qui sont autorisés à utiliser votre service Zimbra. Le référentiel central utilisé pour les données LDAP de Zimbra est le serveur d'annuaire OpenLDAP.

Le serveur LDAP est installé lorsque ZCS est installé. Chaque serveur possède sa propre entrée LDAP qui comprend des attributs spécifiant les paramètres de fonctionnement. En outre, un objet de configuration globale définit par défaut pour tous les serveurs dont l'entrée ne précise pas tous les attributs.

Un sous-ensemble de ces attributs peut être modifié via la console d'administration Zimbra et d'autres via l'utilitaire zmprov CLI.

2.3.1.8. La circulation du trafic LDAP

La figure du trafic d'annuaire LDAP montre le trafic entre le Zimbra-LDAP serveur d'annuaire et les autres serveurs de la Collaboration VMware Zimbra Système serveur. La MTA et Zimbra Collaboration VMware Zimbra Serveur de boîte aux lettres du serveur lue ou écrire à la base de données LDAP sur le serveur d'annuaire.

Les clients Zimbra se connecter via le serveur Zimbra, qui se connecte à LDAP.

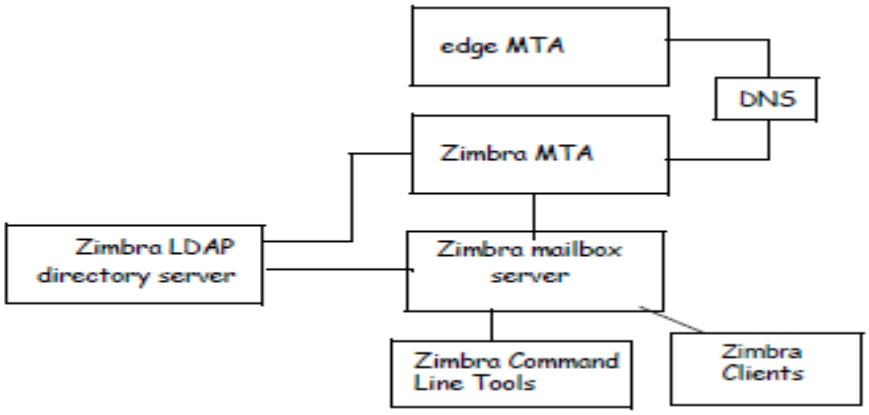


Figure 9: Trafic d'annuaire LDAP

2.3.1.9. Hiérarchie de l'annuaire LDAP

Annuaires LDAP sont disposés en une structure hiérarchique arborescente avec deux types de branches, les branches de messagerie et la branche config. Branches de messagerie sont organisés par domaine. Entrées appartiennent à un domaine, tels que les comptes, des groupes, des alias, sont provisionnés dans le domaine DN dans le répertoire. La branche config contient des entrées de système d'administration qui ne font pas partie d'un domaine. Entrées de la branche config comprennent les comptes du système d'administration, de configuration globale, les subventions globales, COS, les serveurs, les types MIME et Zimlets.

Le chiffre de la hiérarchie LDAP Zimbra montre la hiérarchie LDAP Zimbra. Chaque type d'entrée (objet) a certaines classes d'objets associés.

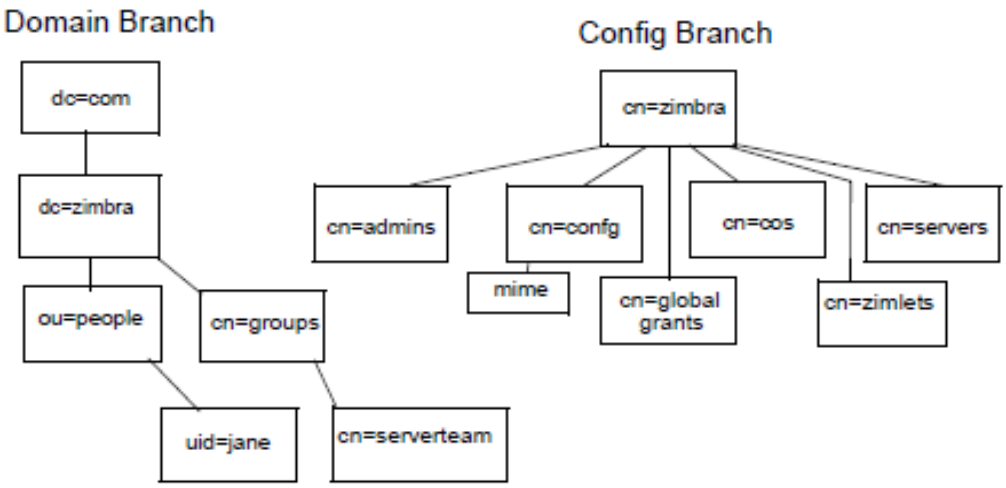


Figure 10: LDAP Zimbra Hiérarchie

Une entrée d'annuaire LDAP se compose d'un ensemble d'attributs et possède un nom distinctif unique au monde (DN). Les attributs souhaités d'une entrée sont déterminées par les classes d'objets associées à cette entrée. Les valeurs des attributs de classe d'objet de déterminer les règles de schéma de l'entrée doit suivre.

La classe d'objet d'une entrée qui détermine le type d'entrée qu'il est, ce qu'on appelle une classe d'objet structurelle et ne peut être modifié. D'autres classes d'objets sont appelés auxiliaires et peuvent être ajoutés ou supprimés à partir de l'entrée.

Utilisation des classes d'objets auxiliaires dans LDAP permet à une classe d'objet doit être combinée avec une classe d'objet existante. Par exemple, une entrée avec structurelle inetOrgPerson classe objet et auxiliaire zimbraAccount de classe d'objet, serait un compte.

2.3.1.10. Le schéma de collaboration VMware Zimbra et serveur LDAP

Au cœur de toutes les implémentations LDAP est une base de données organisée à l'aide d'un schéma.

Le schéma LDAP Zimbra étend le schéma générique inclus avec le logiciel OpenLDAP. Il est conçu pour coexister avec les installations de répertoires existants.

Tous les attributs et classes d'objets créés spécialement pour VMware Zimbra Collaboration Server sont précédées par « Zimbra.», Comme, classe d'objet ou un attribut zimbraAccount zimbraAttachmentsBlocked.

Les fichiers de schéma suivants sont inclus dans la mise en œuvre OpenLDAP:

- core.schema
- cosine.schema
- inetorgperson.schema
- zimbra.schema
- amavisd.schema
- dyngroup.schema
- nis.schema

2.3.1.11. Collaboration du VMware Zimbra et serveur d'objets

Comptes : Représente un compte sur le serveur de messagerie Zimbra qui peut être connecté. Entrées de compte sont des administrateurs ou des comptes d'utilisateurs. Le nom de la classe d'objet est zimbraAccount. Cette classe d'objets étend la classe d'objet zimbraMailRecipient. Tous les comptes ont les propriétés suivantes:

- Un nom dans le format de user@example.domain
- Un identifiant unique qui ne change jamais et n'est jamais réutilisé
- Un ensemble d'attributs, dont certains sont modifiables par l'utilisateur (préférences) et d'autres qui ne sont configurables par les administrateurs
- Tous les comptes utilisateurs sont associés à un domaine, si un domaine doit être créé avant la création des comptes.

Classe d'objet : zimbraAccount

Classe de service (COS) : Définit la valeur par défaut attribue un compte a et quelles fonctionnalités sont autorisés ou refusés. Le COS contrôle les fonctions, les paramètres de préférences par défaut, les quotas de boîtes aux lettres, un message à vie, les restrictions de mot de passe, blocage des pièces jointes et des piscines pour la création de nouveaux comptes serveur.

Classe d'objet : zimbraCOS

Domaine : Représente un domaine de messagerie comme example.com ou example.org. Un domaine doit exister avant que courrier électronique adressé à des utilisateurs dans ce domaine peut être livré.

Classe d'objet : zimbraDomain

Listes de diffusion : Aussi connu sous le nom des listes de diffusion, sont utilisés pour envoyer des messages à tous les membres d'une liste en envoyant un simple email à l'adresse de la liste.

Classe d'objet : zimbraDistributionList

Groupes dynamiques : Sont comme des listes de distribution. La différence est membres d'un groupe dynamique sont calculés de manière dynamique par une recherche LDAP. Le filtre de recherche LDAP est défini dans l'attribut de l'entrée de groupe dynamique.

Remarque: Les listes de distribution et les groupes dynamiques peuvent être utilisés à titre de cessionnaire ou de la cible dans le cadre de l'administrateur délégué.

Classe d'objet : zimbraGroup

Serveurs : Représente un serveur particulier dans le système Zimbra qui a une ou plusieurs des progiciels Zimbra installés. Les attributs décrivent les informations de configuration du serveur, tels que les services qui s'exécutent sur le serveur.

Classe d'objet : zimbraServer

Configuration globale : Indique les valeurs par défaut pour les objets suivants: serveurs et domaines. Si les attributs ne sont pas définies pour d'autres objets, les valeurs sont héritées des paramètres globaux. Les valeurs de configuration globales sont nécessaires et sont définis lors de l'installation dans le cadre du package de base Zimbra. Ceux-ci deviennent les valeurs par défaut pour le système.

Classe d'objet : zimbraGlobalConfig

Alias : Représente un alias d'un compte, la liste de distribution ou d'un groupe dynamique. Les points d'attributs de zimbraAliasTarget de cibler entrée de cette entrée d'alias.

Classe d'objet : zimbraAlias

Zimlet : Définit Zimlets qui sont installés et configurés dans Zimbra.

Classe d'objet : zimbraZimletEntry

Calendrier des ressources : Définit une ressource civile tels que les salles de conférence ou des équipements qui peuvent être sélectionnés pour une réunion. Une ressource de calendrier est un compte avec des attributs supplémentaires sur la classe d'objet zimbraCalendarResource.

Classe d'objet : zimbraCalendarResource

Identité : Représente un personnage d'un utilisateur. Un personnage contient l'identité de l'utilisateur telles que le nom d'affichage et un lien vers l'entrée de signature utilisé pour les emails sortants. Un utilisateur peut créer des personnages multiples. Entrées d'identité sont créées en vertu de l'entrée LDAP de l'utilisateur dans la DIT.
Classe d'objet : zimbraIdentity

Signature : Représente la signature d'un utilisateur. Un utilisateur peut créer plusieurs signatures. Entrées de signature sont créés sous l'entrée LDAP de l'utilisateur dans la DIT.
Classe d'objet : zimbraSignature

2.3.1.12. Mécanisme d'authentification interne

La méthode d'authentification interne utilise le schéma Zimbra cours d'exécution sur le serveur d'annuaire OpenLDAP. Pour les comptes stockés dans le serveur OpenLDAP, l'attribut userPassword stocke un salé-SHA1 (ASIS) digest du mot de passe de l'utilisateur. Fourni le mot de passe de l'utilisateur est calculée dans l'ASIS digest et ensuite comparée à la valeur stockée.

2.3.1.13. Mécanisme d'authentification LDAP et Active Directory externe

L'authentification Active Directory LDAP externe et externe peut être utilisé si l'environnement de messagerie utilise un autre serveur LDAP ou Active Directory de Microsoft pour authentification et Zimbra-LDAP pour tous les autres Collaboration VMware Zimbra Transactions liées au serveur. Cela nécessite que les utilisateurs existent à la fois dans OpenLDAP et dans le serveur LDAP externe.

Les méthodes d'authentification externes tentent de lier au serveur LDAP spécifié serveur en utilisant le nom d'utilisateur et un mot de passe fourni. Si cette liaison réussit, la connexion est fermée et le mot de passe est considéré comme valide.
Les attributs zimbraAuthLdapURL et zimbraAuthLdapBindDn sont nécessaires pour une authentification externe.

- zimbraAuthLdapURL attribut LDAP :/ / ldapserver: port / IP identifie l'adresse ou le nom d'hôte du serveur d'annuaire externe, et est le port nombre. Vous pouvez également utiliser le nom d'hôte complet au lieu du port nombre. Par exemple:
  ldap :/ / server1: 3268
  S'il s'agit d'une connexion SSL, utilisez ldaps : au lieu de ldap:. Le certificat SSL utilisé par le serveur doit être configuré comme un certificat de confiance.
- attribut zimbraAuthLdapBindDn est une chaîne de format utilisée pour déterminer qui DN à utiliser lors de la liaison avec le serveur d'annuaire externe. Au cours du processus d'authentification, le nom d'utilisateur commence dans le format: user@domain.com
  Le nom d'utilisateur peut avoir besoin d'être transformé en un DN de liaison LDAP valide (nom distinctif) dans le répertoire externe.

2.3.1.14. Liste d'adresses globale

La liste d'adresses globale (GAL) est un répertoire d'entreprise des utilisateurs, généralement avec l'organisation elle-même, qui est disponible à tous les utilisateurs du système de messagerie.
VMware Zimbra Collaboration Server utilise l'annuaire d'entreprise pour rechercher des adresses de l'utilisateur au sein de l'entreprise.
Pour chaque domaine de Zimbra Collaboration Server VMware, vous pouvez configurer GAL à utiliser:

- serveur LDAP externe
- serveur LDAP VMware Zimbra Collaboration Server interne

Le serveur Zimbra Collaboration Web Client VMware peut consulter la liste d'adresses globale.

Lorsque l'utilisateur recherche un nom, ce nom est transformé en un filtre de recherche LDAP similaire à l'exemple suivant, où la chaîne% s est le nom que l'utilisateur cherche.***[08]***

- Dans la partie suivante on parle de deux plateformes **Joomla** qui est de la forme CMS et la plateforme **Dokeos** un environnement numérique d'apprentissage. Il s'agit d'une plate-forme d'apprentissage en ligne (ou LMS) comme on a écrit précédemment

2.3.2. Joomla

2.3.2.1.Introduction

Risquons d'abord la métaphore suivante: vous souhaitez construire une nouvelle maison mais vous ne savez trop comment vous y prendre. Vous n'avez pas de connaissances en gros œuvre, en électricité ou encore en décoration, mais vous en avez tellement envie de cette nouvelle maison .Vous pourriez tout apprendre vous-même, enfiler le bleu de travail… et vous tuer à la tâche .Bon, certains y arrivent, c'est vrai.

Vous avez donc pris contact avec plusieurs maîtres d'œuvre et l'un d'eux a particulièrement retenu votre attention: il s'occupe du gros œuvre et vous livre une maison modulable où vous pourrez choisir vous-même l'emplacement des cloisons (pour faire autant de pièces que vous voulez), et la décoration. Il ne vous reste plus qu'à meubler.

La maison c'est votre site Web, le maître d'œuvre c'est Joomla, le gros œuvre c'est l'environnement de travail PHP/MySQL, les cloisons c'est précisément la modularité de Joomla (qui vous permettra notamment d'ajouter des composants et des modules à la structure de l'édifice), la décoration c'est le template (le design de votre site), quant aux meubles, vous l'aurez deviné, il s'agit du contenu même de votre site.

2.3.2.2. Système de gestion de contenu

Un CMS (système de gestion de contenu) est un logiciel web qui permet de créer un site Internet dynamique en toute simplicité, sans connaissances techniques particulières, l'idée étant de séparer la forme du contenu : vous saisissez un article et Joomla! S'occupe de le publier au bon endroit avec la bonne mise en page

2.3.2.3. Définition de Joomla

Joomla est un outil de gestion de contenu (en anglais, CMS, pour Content Management system) Open Source sous licence GNU/GPL créé par une équipe internationale de développeurs récompensée à maintes reprises.

2.3.2.4.La mise en œuvre de Joomla

Avant de se lancer dans la mise en œuvre de Joomla et sa configuration, nous avons préféré vous présenter Joomla avec son vocabulaire, les concepts de base et quelques exemples.

2.3.2.5. Les notions de base

2.3.2.5.1. La terminologie Joomla

Voici une liste des termes les plus fréquemment utilisés dans la planète Joomla, qui vous aidera à mieux appréhender son fonctionnement:

Article : un article est une unité de contenu. Il comprend généralement du texte, des images et des liens ; il a certaines caractéristiques comme un titre, un auteur, une date de publication et tout un tas de paramètres qui seront décrits plus loin.

> Un article est placé dans une rubrique, elle-même fait partie d'une section. Mais il existe des articles non catégorisé – appelés articles statiques dans les précédentes versions de Joomla.

Menu : c'est une liste d'éléments, disposés de façon verticale ou horizontale selon le module choisi pour l'afficher et sa configuration. L'appui sur un élément du menu provoque l'affichage d'une page avec ses modules et ses composants …

Page d'accueil : c'est la première page que voit un visiteur lorsqu'il saisit le nom de votre site.

Administration : la partie administration – ou backend - est l'arrière-boutique de votre site ; l'interface d'administration va permettre de créer et mettre à jour vos articles mais aussi de gérer tout votre site.

Site : La partie Site - ou frontend - c'est la boutique, ce que voient les visiteurs qui viennent sur votre site.

Cache : pour rendre plus rapide l'affichage des pages de votre site, les éléments les plus souvent demandés (logos, images, page d'accueil) sont stockés dans un répertoire intermédiaire, encore appelé cache. Lorsqu'un utilisateur veut consulter une page comprenant un élément en cache, Joomla n'a plus besoin d'aller le chercher dans la base de données ou un répertoire du site, il le prend directement dans le cache.

> Le cache est mis à jour régulièrement, mais si vous avez fait des mises à jour importantes de votre site, il vaut mieux nettoyer votre cache, c'est à dire supprimer tous les fichiers mis dans le cache, au travers de l'interface d'administration.

Core team (CT): la Core Team est l'équipe de bénévoles en charge du développement du code source et de l'organisation générale du projet Joomla! Elle est à ce jour composée d'une quinzaine de membres (développeurs et anglophones pour l'essentiel).

Editeur WYSIWYG : comme son nom l'indique, il s'agit d'un éditeur qui va permettre de rédiger et de mettre en forme du texte comme vous le feriez avec un traitement de texte (What You See Is What You Get), sans vous soucier du code html sous-jacent.

Publier / **dépublier** : encore une notion importante à intégrer. Il s'agit de rendre visible ou pas sur le site un article, un lien dans un menu, un module entier, une section, une catégorie, un article. Pour un article, il est par ailleurs possible de définir un calendrier de publication, date à partir de laquelle ou jusqu'à laquelle un article sera publié.

2.3.2.5.2. Les extensions

Joomla est un outil de gestion de contenu assez sophistiqué qui s'appuie sur des extensions, c'est-à-dire des programmes complémentaires pour gérer la mise en forme ou ajouter des nouveaux services. Ces extensions sont classées en 4 catégories : les composants, les modules, les plugins et les templates. La version standard de Joomla intègre un certain nombre d'extensions mais vous en trouverez quelques milliers sur le net pour personnaliser votre site

Composant : c'est une mini application intégrée à votre site Joomla, qui dispose de sa propre interface de configuration dans la console d'administration Joomla.

Ainsi à chaque fois qu'une page est chargée, Joomla fait appel à un composant pour générer le corps de la page ; de même, il existe un composant pour authentifier les utilisateurs … Les composants constituent la majeure partie de vos pages ! Les composants de base sont fournis avec Joomla. D'autres composants peuvent être facilement installés par la suite (forums, livre d'or, galerie d'images, gestionnaire de newsletter, gestionnaire de formulaires… et bien d'autres encore).

Exemple : com_content (gestion des contenus) et *com_registration* (enregistrement des utilisateurs)

Module : pour faire simple, un module est un bloc que l'on trouvera généralement autour du corps de la page web, par exemple dans la colonne de gauche ou la colonne de droite de notre site. Ainsi le menu de gauche de votre site est placé dans un module ! De même que la bannière en haut de votre site, le bas de page ou le module d'identification …

Les modules sont souvent associés à des composants, comme par exemple le module qui affiche une photo aléatoire tiré d'une galerie d'images géré par un composant.

Exemple : mod_banners (affichage des bannières*), mod_mainmenu* (affichage d'un menu)

Plugin : ce sont des morceaux de code activés sur un évènement. L'exécution de n'importe quelle partie de Joomla, que ce soit le noyau, un module ou un composant, peut déclencher un évènement et alors les plugins associés à cet événement s'exécuteront. ***[06]***

Un plug-in ajoute des capacités spécifiques à un composant. Le terme plug-in est également utilisé à d'autres endroits. Par exemple, les plug-ins sont communément utilisés dans les navigateurs web pour lire les vidéos. Un exemple de plug-in bien connu est Adobe's Flash Player. Un bon exemple de l'utilisation de plug-ins dans Joomla est le Composant de recherche. Cinq plug-ins de recherche travaillent ensemble pour trouver le contenu venant de différents composants de Joomla. Celui-ci dispose huit types de plugin:

Authentification, captcha, content, editors-xtd, editors, extension, finder, quickicon, search, system et *user*. Ce sont également les noms des sous-répertoires dans lesquels sont rangés les fichiers de ces plug-ins. Par exemple, les plug-ins de type *authentification* sont localisés dans le répertoire *plugins/authentification*. Il n'est pas possible ni nécessaire de créer un plug-in dans la zone administration comme nous l'avons vu dans le chapitre modules. Un plug-in doit être installé via le Gestionnaire d'extensions.

Authentification

L'autorisation est le processus de spécification des droits d'accès. Il est précédé par l'authentification, qui vérifie si la personne qui essaye d'être autorisée fournit des informations d'identification correctes.

Vous vous authentifiez avec votre identifiant et votre mot de passe, et vous êtes autorisé parce que vous êtes un membre d'un groupe possédant les autorisations.

Joomla offre trois possibilités pour l'authentification. Soyez prudent avec la désactivation des plug-ins. Vous devez avoir au moins un plug-in d'authentification activé ou vous perdrez tout accès à votre site.

Figure 11: plug-ins Authentification

Joomla

Le plug-in fournit le comportement standard pour Joomla. Vous remplissez le formulaire de connexion avec votre identifiant et votre mot de passe, puis vos informations de connexion sont ensuite vérifiées.

GMail

Si vous activez le plug-in Gmail, les utilisateurs pourront se connecter au site en utilisant leur adresse Gmail et leur mot de passe. L'enregistrement préalable n'est pas nécessaire. Avec la première connexion le *System plug-in Joomla*va créer un compte utilisateur dans la base de données. Le mot de passe Gmail est stocké en crypté dans la base de données, afin que vos utilisateurs se connectant avec leurs comptes Gmail ne puissent pas être piratés. Ce Plug-in facilite le processus de connexion pour vos utilisateurs. Malheureusement, il n'y a pas d'indication dans le formulaire de connexion expliquant qu'il est possible de s'identifier avec Gmail. Vous devrez ajouter du texte supplémentaire ou imaginer une solution alternative.

LDAP

Le *Lightweight Directory Access Protocol (LDAP)* est un protocole d'application pour la lecture et l'édition des données des services d'annuaire. C'est utilisé dans les sociétés pour l'affiliation des départements de gestion ainsi que pour les numéros de téléphone des employés.

dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Barbara Doe,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top

Pour pouvoir utiliser ce Plug-in pour l'authentification, vous avez besoin d'un serveur LDAP (Open LDAP) et vous devez configurer le Plug-in LDAP avec les données spécifiques du serveur. Vous trouverez un bon tutoriel sur joomla.org: *LDAP from Scratch*.***[05]***

Template : un template gère toute la partie graphique de votre site : les couleurs des caractères et des fonds, la police des caractères, les cadres, les menus ... En changeant de template, vous changez le « look and feel » de votre site.

Il en existe des centaines, disponibles gratuitement sur des sites qui se sont spécialisés dans cette activité. Nous verrons plus loin comment installer et personnaliser un template. Le terme template est l'équivalent de skin, thème ou encore gabarit dans d'autres CMS. Les templates proposés par défaut avec Joomla sont rhuk milkyway, beez, et JA Purity … [06]

2.3.3. Dokeos

2.3.3.1. Gestion de système d'apprentissage

Un learning management system (LMS) ou learning support system (LSS) est un système logiciel web développé pour accompagner toute personne impliquée dans un processus d'apprentissage dans sa gestion de parcours pédagogiques. Les services offerts incluent généralement un contrôle d'accès, des outils de communication (synchrones et/ou asynchrones) et l'administration des groupes d'utilisateurs. En français, on trouve les appellations : plate-forme d'apprentissage en ligne, Système de gestion de l'apprentissage, centre de formation virtuel, plate-forme e-learning (FOAD).

Le système informatique mis en place du côté serveur est appelé CMS (content management system) ou un ENT (espace numérique de travail). Des fonctionnalités peuvent leurs être associés en fonction du cahier des charges.***[17]***

2.3.3.2. Définition de Dokeos

Dokeos est une plate-forme d'apprentissage à distance (ou plate-forme d'e-learning).
D'une grande simplicité de mise en œuvre et très intuitive pour ses utilisateurs (formateurs, stagiaires, auditeurs de la formation continue, etc...), ***Dokeos*** propose de nombreux outils destinés à organiser les apprentissages et laisse toute latitude à votre créativité pour élaborer des cours réellement attractifs, interactifs et multimédias. ***Dokeos*** met aussi à la disposition des utilisateurs des outils de travail collaboratif : forums, blog, wiki… Outre cette simplicité d'utilisation, ***Dokeos*** présente l'avantage non négligeable d'être un logiciel libre, dont le code source est accessible et peut être modifié ou adapté pour des besoins plus spécifiques.

2.3.3.3. L'utilisation du *Dokeos* :

Dokeos regroupe, sous une interface commune :

• un environnement personnel d'apprentissage (PLE) performant et ergonomique

• des outils de conception de contenu en ligne :

o création rapide de contenu avec ou sans modèles

o création de tests et d'enquêtes

o conversion de présentations en cours

o importation de cours conformes au standard SCORM

• des outils d'apprentissage collaboratifs :

o forum de discussion

o wiki

o blog

• des outils de suivi (reporting) avancé permettant de mesurer les progrès des utilisateurs :

o temps passé dans les cours

o résultats des tests et enquêtes

o export des données vers un tableur ***[03]***

2.4. Conclusion

Dans ce chapitre on a mets la lumière sur les outils qu'on a utilisé dans notre travail l'authentification unifiée les plateformes et le serveur LDAP qui est le point de base pour et relier les autre plateformes pour un seul login dans le chapitre suivant on présente l'implémentation et l'installation de chacun des plateformes.

Chapitre IV :L'implémentation

1. Introduction

Parmi les étapes les plus importantes de l'implémentation l'étape de conception, qu'on ne peut pas y dépasse et sans passer de cet étape on trouve des grandes erreurs dans les travails Pour la conception des programmes plusieurs outils peuvent être utilisés, parmi ces outils le langage UML qui est choisi pour la conception de notre programme

Parmi les multiples outils logiciels utilisés pour dessiner nos diagrammes, nous avons choisi StarUML par ce qu'il est libre et gratuit et supporte la version 2.0 d'UML.

On a présenté aussi l'installation et la mise en œuvre de différentes plateformes et les étapes de configuration pour obtenir l'authentification uniques entre les différentes plateformes.

Dans notre travail on utilise deux méthodes différentes :

3. Conception :

Dans ce qui suit, on va présenter la conception (les différents diagrammes) de notre projet en utilisant UML. On fait la présentation de diagramme de cas d'utilisation générale.

2.1. Modélisation d'authentification

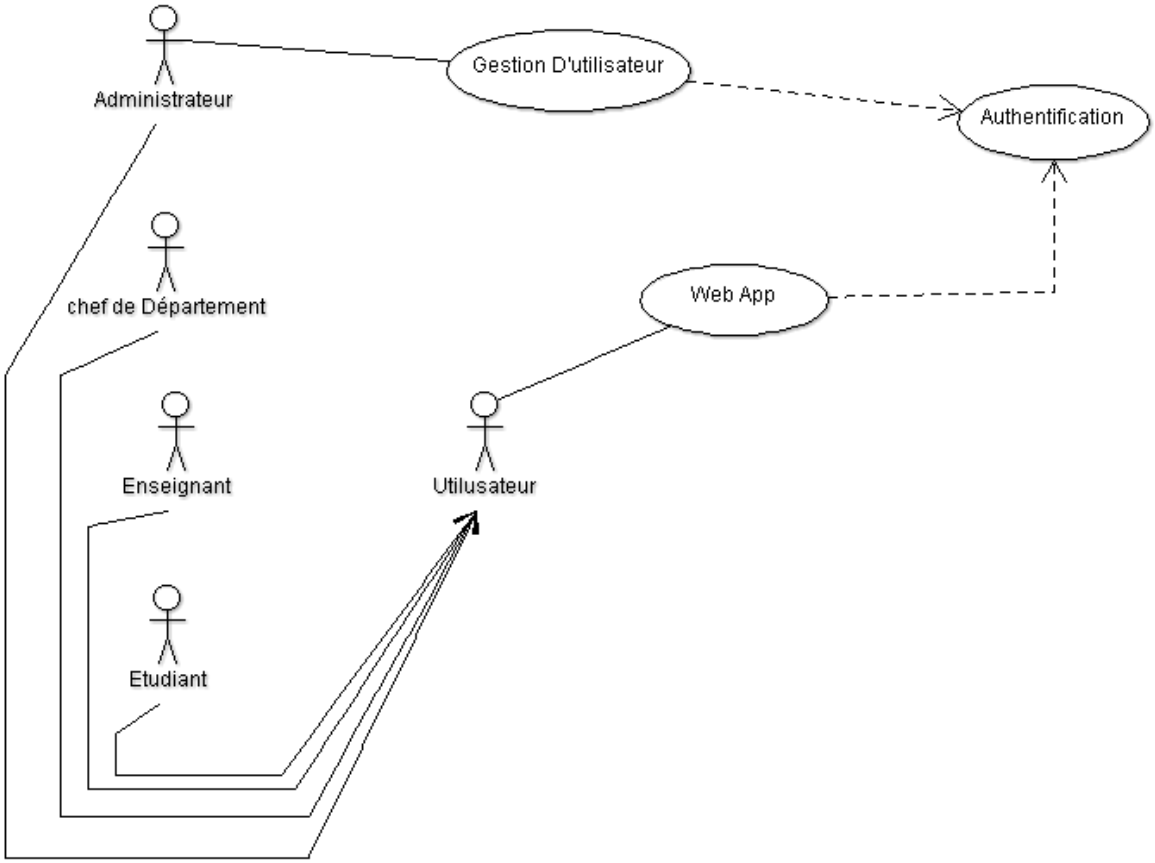


Figure 12: Diagramme de cas d'utilisation

Description	
Titre	Authentification.
But	Ce cas d'utilisation permet à un utilisateur de se connecter aux Plateformes
Acteurs	Utilisateur
Description des enchainements	
Enchainements	L'utilisateur doit être un chef de département ou un enseignant ou un étudiant ou un administrateur , ils peuvent accédés aux applications web avec l'authentification. L'administrateur qui fait la gestion des utilisateurs pour qu'ils puissent authentifier et accéder au application web.

Tableau 2: description diagramme des cas d'utilisation

Cas d'utilisation Authentification :

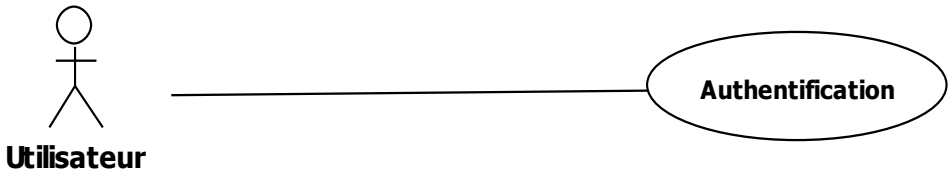


Figure 13 : Cas d'utilisation d'authentification

3.1. Diagramme de séquence de cas authentification :

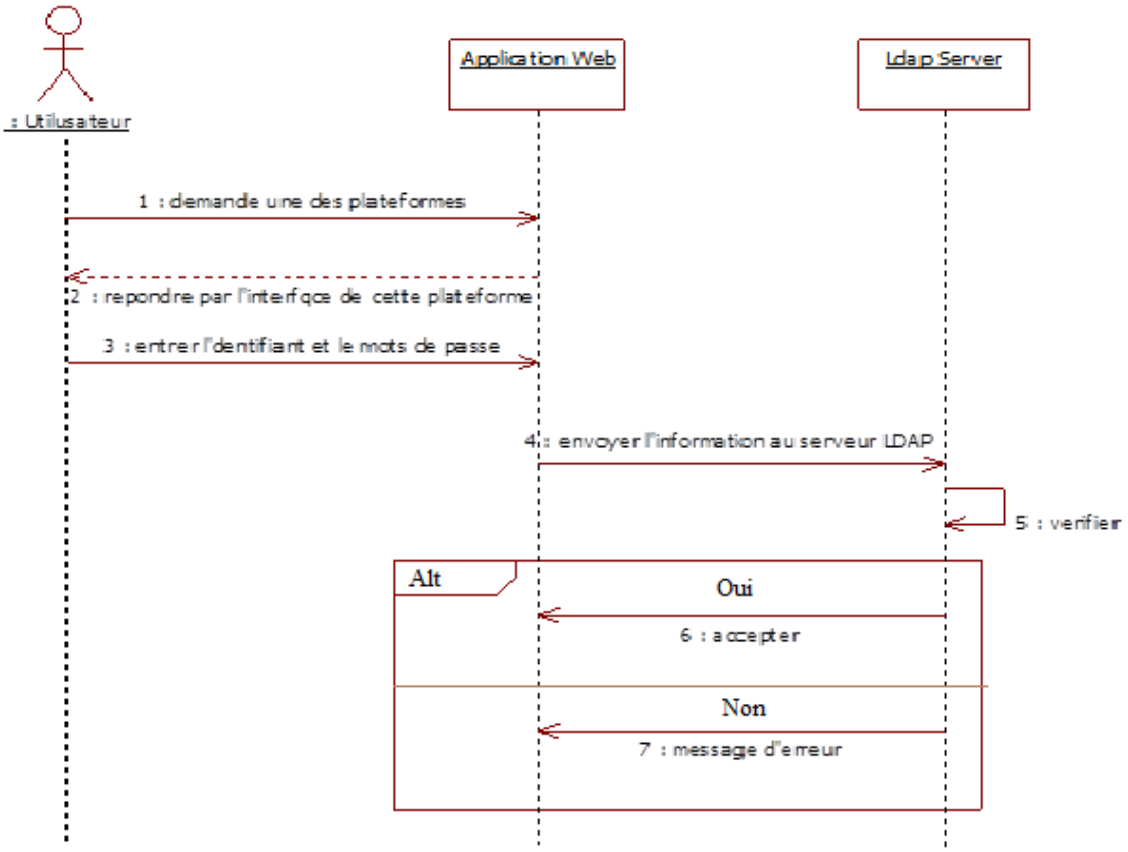


Figure 14: Diagramme de séquence de cas authentification

Descriptions des cas d'authentification :

Description	
Titre	Diagramme de séquence de cas authentification
But	Ce cas d'utilisation permet à un utilisateur de se connecter au Plateforme
Acteurs	Utilisateur
Description des enchainements	
Enchainements	1- L'utilisateur demande une des plateformes . 2- Répondre par l'interface de cette plateforme. 3- Entrer l'identifiant et le mot de passe . 4- Envoyer l'information au serveur LDAP . 5- Vérification . 6- Si oui LDAP l'accepte, on peut accéder au plateforme 7- Si non , affichage d'un message d'erreur . Exception : dans le cas où l'utilisateur rentre un login et/ou un mot de passe erroné: la plateforme donne un message d'erreur

Tableau 3: description des cas d'authentification

4. L'installation du open LDAP

4.1. Installation initial:

Installer les paquets nécessaires

```
[root@dir ~]# yum -y install openldap-servers openldap-clients
```

Activation de la prise en charge de LDAP par le serveur LDAP

Si l'on veut pouvoir interroger le serveur en LDAPS, il faudra éditer /etc/sysconfig/ldap

```
[root@dir ~]# vi /etc/sysconfig/ldap
SLAPD_LDAPI=Yes
```

Editer le fichier de configuration slapd.conf avec vim .

Le fichier slapd.conf, qui se trouve dans /etc/openldap, contient les informations de configuration nécessaires à votre serveur LDAP **slapd**. Il vous faudra éditer ce fichier pour le rendre spécifique à vos domaines et serveur.

```
[root@dir ~]# vi /etc/openldap/slapd.conf
```

slapd.conf : ce fichier comporte diverses informations telles que la racine supérieure de l'annuaire, l'administrateur principal de l'annuaire LDAP et son mot de passe, les droits d'accès par défaut, les fichiers d'objets et de syntaxe à utiliser ainsi que les règles d'accès pour les entrées et les attributs de l'annuaire LDAP.

```
# create new

pidfile     /var/run/openldap/slapd.pid
argsfile    /var/run/openldap/slapd.args
```

Effacer le repertoire slapd.d (sinon CentOS ne pourra pas prendre en compte notre

```
[root@dir ~]# rm -rf /etc/openldap/slapd.d/*
```

slaptest : Teste la validité du fichier de configuration slapd.conf

convertir le repertoire slapd.conf en un fichier slapd.d

```
[root@dir ~]# slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d clients
```

A l'installation du Openldap, l'installateur va créer par défaut une base, nous les éditer par vim .

```
[root@dir ~]# vi /etc/openldap/slapd.d/cn=config/olcDatabase\={0}config.ldif
```

{0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth manage by * break

```
[root@dir ~]# vi /etc/openldap/slapd.d/cn=config/olcDatabase\={1}monitor.ldif
```

```
# create new

dn: olcDatabase={1}monitor
objectClass: olcDatabaseConfig
olcDatabase: {1}monitor
olcAccess: {1}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
manage by * break
olcAddContentAcl: FALSE
olcLastMod: TRUE
olcMaxDerefDepth: 15
olcReadOnly: FALSE
olcMonitoring: FALSE
structuralObjectClass: olcDatabaseConfig
creatorsName: cn=config
modifiersName: cn=config



[root@dir ~]# chown -R ldap. /etc/openldap/slapd.d


[root@dir ~]# chmod -R 700 /etc/openldap/slapd.d


[root@dir ~]# /etc/rc.d/init.d/slapd start


Starting slapd: [ OK ]
```

Configure le lancement automatique a chaque redémarrage du système

```
[root@dir ~]# chkconfig slapd on
```

4.2. La configuration initial

On ajoute les schémas utilisés

Ajout des paramètres de la base à partir du fichier préparé ci-dessus

```
[root@dir ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/core.ldif

SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=core,cn=schema,cn=config"
[root@dir ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif

SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=cosine,cn=schema,cn=config
[root@dir ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif


SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=nis,cn=schema,cn=config"

[root@dir ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif


SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=inetorgperson,cn=schema,cn=config"
```

On code tout d'abord le mot de passe root de la base

```
[root@dir ~]# slappasswd

New password:

Re-enter new password:
{SSHA}xxxxxxxxxxxxxxxxxxxxxxxx

[root@dir ~]# vi backend.ldif
```

Créer un nouveau

On remplace la section "dc = ***, dc = ***" avec notre propre suffixe

On remplace la section "olcRootPW: ***" pour notre propre mot de passe généré par slappasswd dessus

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib64/openldap
olcModuleload: back_hdb
dn: olcDatabase=hdb,cn=config

objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcSuffix: dc=ukmo,dc=dz
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin,dc=ukmo,dc=dz
olcRootPW: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxx
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_lk_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcMonitoring: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=server,dc=world" write by anonymous auth
by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=server,dc=world" write by * read

[root@dir ~]# ldapadd -Y EXTERNAL -H ldapi:/// -f backend.ldif
SASL/EXTERNAL authentication started
SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
SASL SSF: 0
adding new entry "cn=module,cn=config"

adding new entry "olcDatabase=hdb,cn=config"

[root@dir ~]# vi frontend.ldif
```

Créer un nouveau

On remplace la section "dc = ***, dc = ***" avec notre propre suffixe

On remplace la section "olcRootPW: ***" pour notre propre mot de passe généré par slappasswd dessus

```
dn: dc=ukmo,dc=dz
objectClass: top
objectClass: dcObject
objectclass: organization
o: Server World
dc: Server

dn: cn=admin,dc=ukmo,dc=dz
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
userPassword: {SSHA}xxxxxxxxxxxxxxxxxxxxxxxx

dn: ou=people,dc=ukmo,dc=dz
objectClass: organizationalUnit
ou: people

dn: ou=groups,dc=ukmo,dc=dz
objectClass: organizationalUnit
ou: groups

[root@dir ~]# ldapadd -x -D cn=admin,dc=server,dc=world -W -f frontend.ldif


Enter LDAP Password:

ajouter des  nouvelles entrées

adding new entry "dc=ukmo,dc=dz"

adding new entry "cn=admin,dc= ukmo,dc= dz "

adding new entry "ou=people,dc= ukmo,dc= dz "

adding new entry "ou=groups,dc=ukmo,dc= dz "
```

4.3. Ajouter des utilisateurs locaux existants de l'annuaire LDAP

```
[root@dir ~]# vi ldapuser.sh
```

Extraire les utilisateurs locaux qui ont 500-999 UID chiffres

Remplacer "suffix = ***" pour votre propre suffixe

```
#!/bin/bash

SUFFIX='dc=ukmo,dc=dz'
LDIF='ldapuser.ldif'

echo -n > $LDIFfor line in `grep "x:[5-9][0-9][0-9]:" /etc/passwd | sed -e "s/ /%/g"`
do
  UID1=`echo $line | cut -d: -f1`
  NAME=`echo $line | cut -d: -f5 | cut -d, -f1`
  if [ ! "$NAME" ]
  then
    NAME=$UID1
  else
    NAME=`echo $NAME | sed -e "s/%/ /g"`
  fi
  SN=`echo $NAME | awk '{print $2}'`
  if [ ! "$SN" ]
  then
    SN=$NAME
  fi
  GIVEN=`echo $NAME | awk '{print $1}'`
  UID2=`echo $line | cut -d: -f3`
  GID=`echo $line | cut -d: -f4`
  PASS=`grep $UID1: /etc/shadow | cut -d: -f2`
  SHELL=`echo $line | cut -d: -f7`
  HOME=`echo $line | cut -d: -f6`
  EXPIRE=`passwd -S $UID1 | awk '{print $7}'`
  FLAG=`grep $UID1: /etc/shadow | cut -d: -f9`
```

```
if [ ! "$FLAG" ]
  then
    FLAG="0"
  fi
  WARN=`passwd -S $UID1 | awk '{print $6}'`
  MIN=`passwd -S $UID1 | awk '{print $4}'`
  MAX=`passwd -S $UID1 | awk '{print $5}'`
  LAST=`grep $UID1: /etc/shadow | cut -d: -f3`

  echo "dn: uid=$UID1,ou=people,$SUFFIX" >> $LDIF
  echo "objectClass: inetOrgPerson" >> $LDIF
  echo "objectClass: posixAccount" >> $LDIF
  echo "objectClass: shadowAccount" >> $LDIF
  echo "uid: $UID1" >> $LDIF
  echo "sn: $SN" >> $LDIF
  echo "givenName: $GIVEN" >> $LDIF
  echo "cn: $NAME" >> $LDIF
  echo "displayName: $NAME" >> $LDIF
  echo "uidNumber: $UID2" >> $LDIF
  echo "gidNumber: $GID" >> $LDIF
  echo "userPassword: {crypt}$PASS" >> $LDIF
  echo "gecos: $NAME" >> $LDIF
  echo "loginShell: $SHELL" >> $LDIF
  echo "homeDirectory: $HOME" >> $LDIF
  echo "shadowExpire: $EXPIRE" >> $LDIF
  echo "shadowFlag: $FLAG" >> $LDIF
  echo "shadowWarning: $WARN" >> $LDIF
  echo "shadowMin: $MIN" >> $LDIF
  echo "shadowMax: $MAX" >> $LDIF
  echo "shadowLastChange: $LAST" >> $LDIF
  echo >> $LDIF
done
```

```
[root@dir ~]# sh ldapuser.sh

[root@dir ~]# ldapadd -x -D cn=admin,dc=ukmo,dc=dz -W -f ldapuser.ldif


Enter LDAP Password:

adding new entry "uid=cent,ou=people,dc=ukmo,dc=dz"

adding new entry "uid=fedora,ou=people,dc= ukmo,dc=dz"

adding new entry "uid=ubuntu,ou=people,dc= ukmo,dc=dz"

adding new entry "uid=debian,ou=people,dc= ukmo,dc=dz"
```

4.4. Ajouter des groupes locaux existants dans le répertoire LDAP

```
[root@dir ~]# vi ldapgroup.sh
```

Extrait groupes locaux qui ont 500-999 UID chiffres

Remplacer "suffix = ***" pour votre propre suffixe

```
#!/bin/bash

SUFFIX='dc=ukmo,dc=dz'
LDIF='ldapgroup.ldif'

echo -n > $LDIF
for line in `grep "x:[5-9][0-9][0-9]:" /etc/group`
do
  CN=`echo $line | cut -d: -f1`
  GID=`echo $line | cut -d: -f3`
  echo "dn: cn=$CN,ou=groups,$SUFFIX" >> $LDIF
  echo "objectClass: posixGroup" >> $LDIF
  echo "cn: $CN" >> $LDIF
  echo "gidNumber: $GID" >> $LDIF
  users=`echo $line | cut -d: -f4 | sed "s/,/ /g"`
  for user in ${users} ; do
    echo "memberUid: ${user}" >> $LDIF
  done
  echo >> $LDIF
done
[root@dir ~]# sh ldapgroup.sh

[root@dir ~]# ldapadd -x -D cn=admin,dc=ukmo,dc=dz -W -f ldapgroup.ldif
```

[13]

```
Enter LDAP Password:

adding new entry "cn=cent,ou=groups,dc=ukmo,dc=dz"

adding new entry "cn=fedora,ou=groups,dc= ukmo,dc=dz"

adding new entry "cn=ubuntu,ou=groups,dc= ukmo,dc=dz"

adding new entry "cn=debian,ou=groups,dc= ukmo,dc=dz"

adding new entry "cn=fermi,ou=groups,dc=ukmo,dc=dz"
```

5. Configuration DNS

Installer et configurer BIND

```
[root@dlp ~]# yum -y install bind bind-utils
```

named.conf c'est de le paquet de CentOS pour configure le ISC BIND de serveur DNS comme un localhost un cache nom de serveur

```
[root@dlp ~]# vi /etc/named.conf
```

La configuration principale du DNS ressemblera ci-dessous. On Modifier et on ajouter les entrées et on définit les zones.

```
zone "." IN {
                type hint;
                file "named.ca";
        };
        zone "ukmo.dz" IN {
                type master;
          file "ukmo.dz.zone";
                allow-update { none; };
        };
        zone "1.168.192.in-addr.arpa" IN {
                type master;
                file "1.168.192.zone";
                allow-update { none; };
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

On fait une copier pour la zone on cas de la perte de ce dernier

```
Bind # cp db.local ukmo.dz.zone
```

Début BIND. *[20]*

```
[root@dlp ~]# /etc/rc.d/init.d/named start

Starting named:  [  OK  ]

[root@dlp ~]# chkconfig named on
```

Modifier les paramètres où le serveur de résolution de noms.

```
[root@dlp ~]# vi /etc/resolv.conf

search  ukmo.dz
nameserver  192.168.1.2
```

On peut changer le nom de hostname

```
# hostname ns.ukmo.dz
```

On assure que serveur peut résoudre les noms de domaine ou des adresses IP.

```
[root@dlp ~]# dig ns.ukmo.dz.
```

```
# service Bind start

# ns lookup
```

```
[root@dlp ~]# vi /var/named/ukmo.dz

@          IN          NS          ns.ukmo.dz.
Ns         IN          A           192.168.1.2.
@          IN          MX10        mail.ukmo.dz.
Mail       IN          A           192.168.1.4
Dokeos     IN          A           192.168.1.5
Moodle     IN          A           192.168.1.6
Joomla     IN          A           192.168.1.7
```

Configurer DNS que comme serveur esclave. Il est facile de le mettre en place.

Un environnement DNS maître est "ns.ukmo.dz".

on écrit config dans le dossier de la zone de DNS maître.*[12]*

6. L'installation du Zimbra

Télécharger zimbra

1. Se rendre sur le site du projet, ou plus simplement utiliser wget.

wgethttp://files2.zimbra.com/downloads/8.0.2_GA/zcs8.0.2_GA_5569.RHEL6_64.20121210115059.tgz

Il s'agit de la version disponible au moment de la rédaction du billet

Il est important d'avoir une configuration correcte DNS notamment au niveau d'un enregistrement MX.

2. Décompacter les archives tar

tarzxvf zcs-8.0.2_GA_5569.RHEL6_64.20121210115059.tgz

3. Se positionner dans le répertoire d'installation de zimbra:

cdzcs-8.0.2_GA_5569.RHEL6_64.20121210115059

[root@mail]# zcs-8.0.2_GA_5569.RHEL6_64.20121210115059

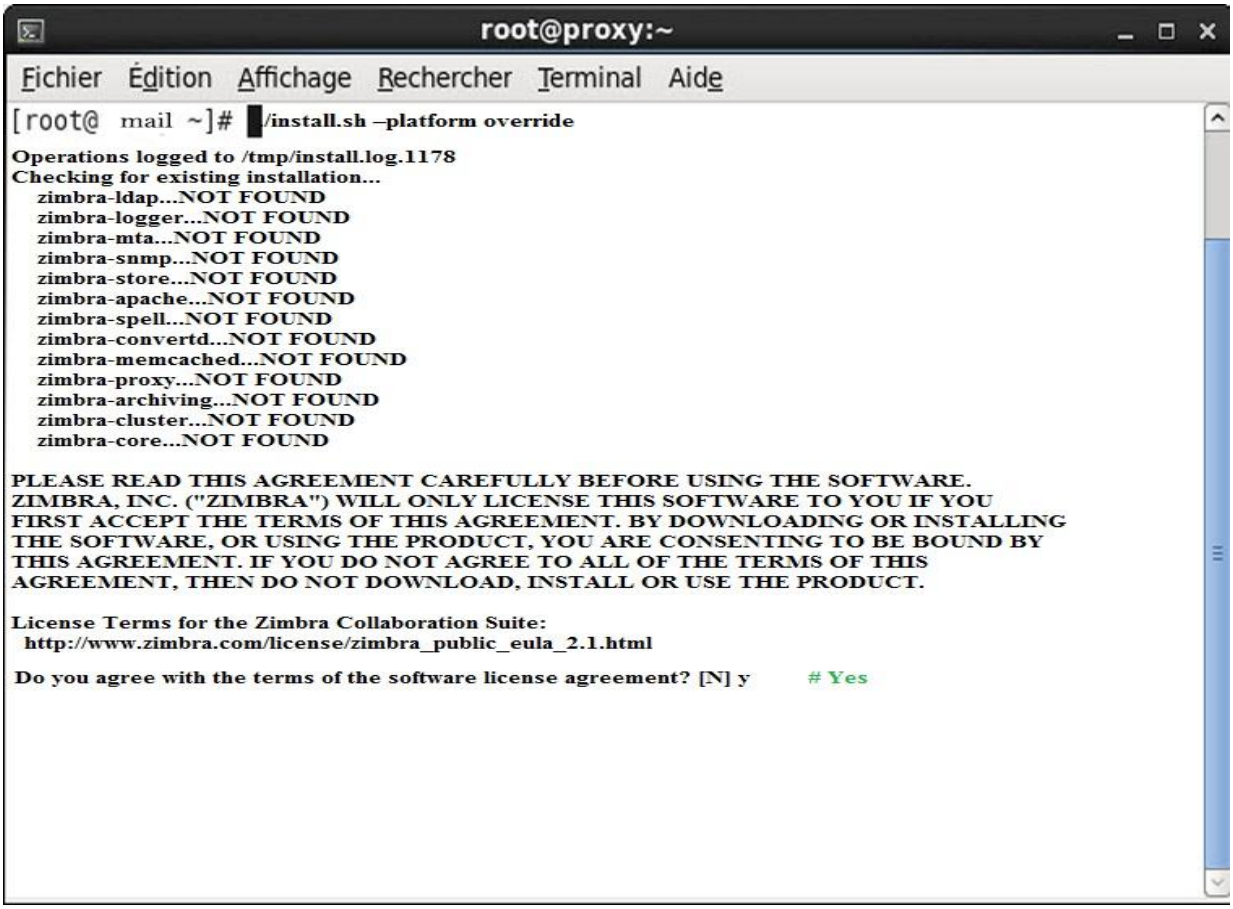


Figure 15: la 1er commande pour lance l'installation de Zimbra

A cette figure : l'installation de plateforme et si nous avons un paquet qui manqué, nous devons l'installer manuellement et Lecture de la Licence on répond par Yes

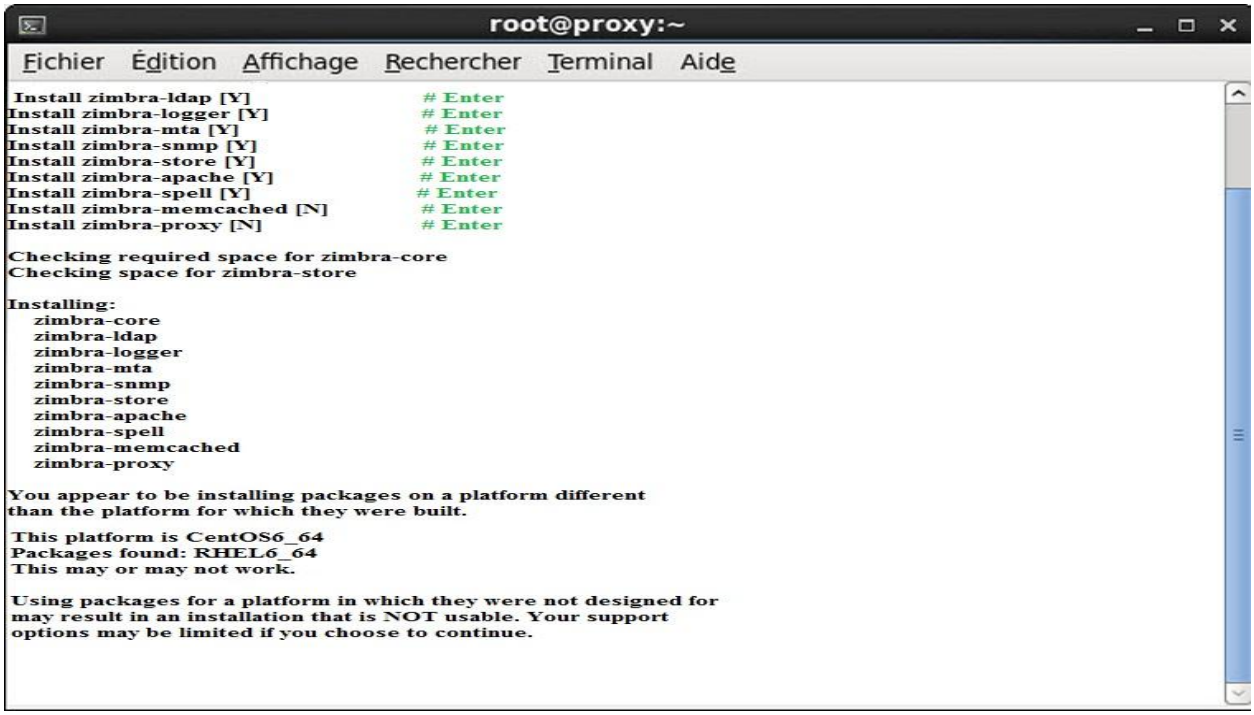


Figure 16: Vérification des packages Zimbra core

Vérification des packages de Zimbra store (core) ensuite leur installation

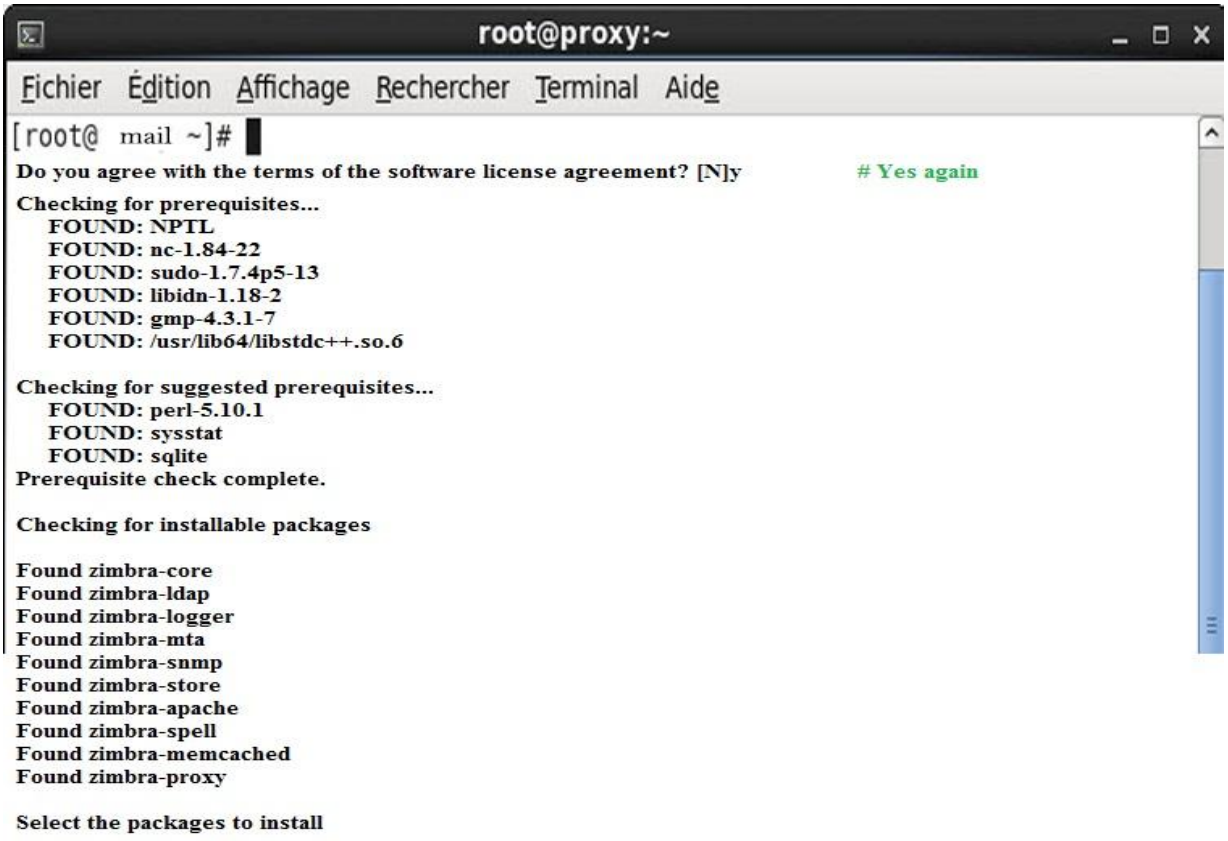


Figure 17: : Lecture de License

La lecture de licence et la vérification des paquets de stockage de Zimbra et Vérification desconditions préalables

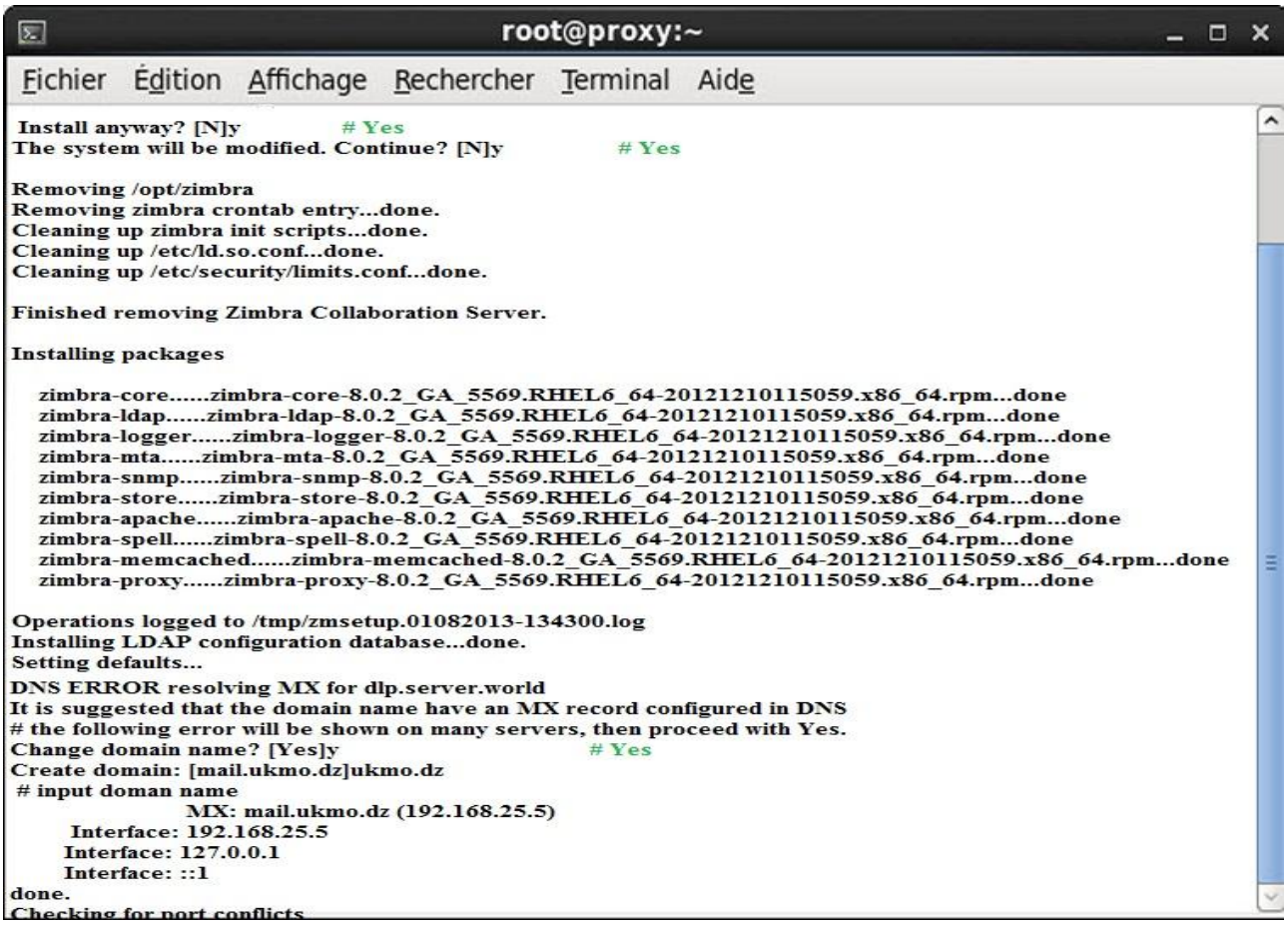


Figure 18: donner le domaine de messagerie électronique et IP

L'installation après la vérification des paquets et l'intégration de domaine de messagerie électronique et leur IP.

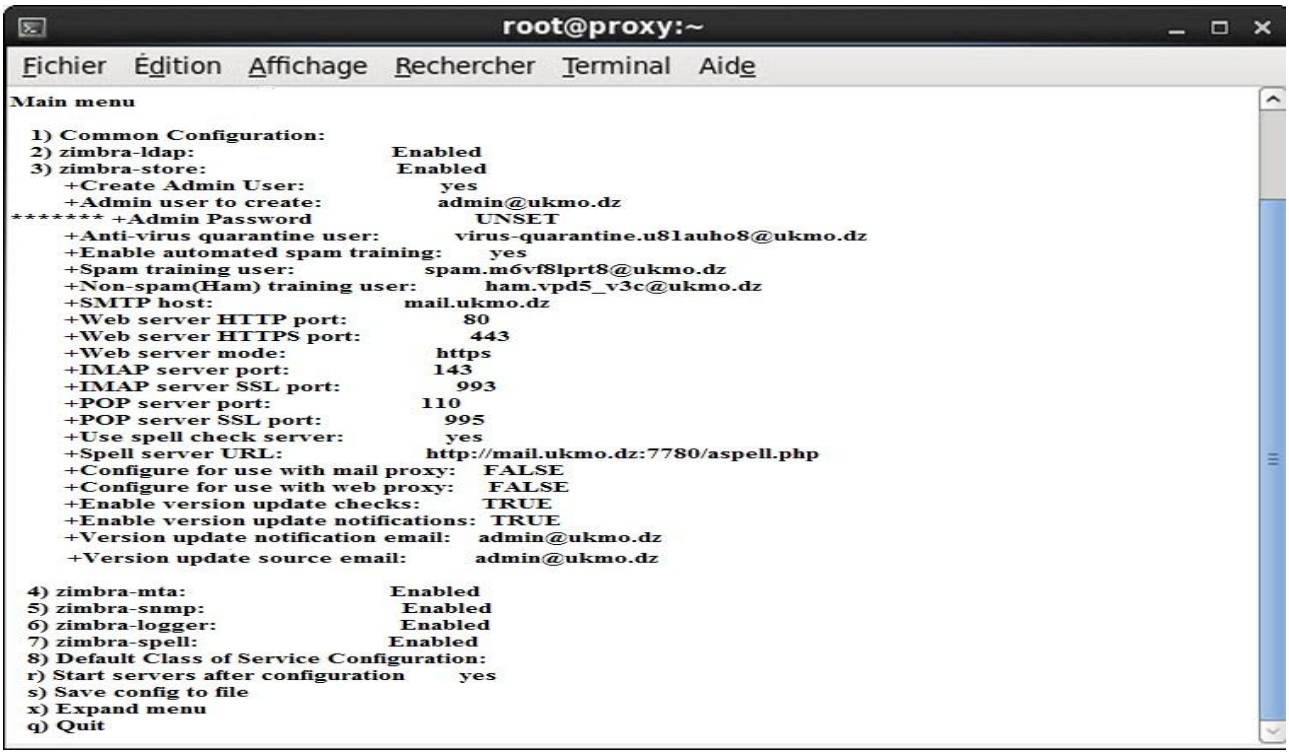


Figure 19:l'affichage de menu principal

Vérification des conflits de ports et l'affichage de menu principal

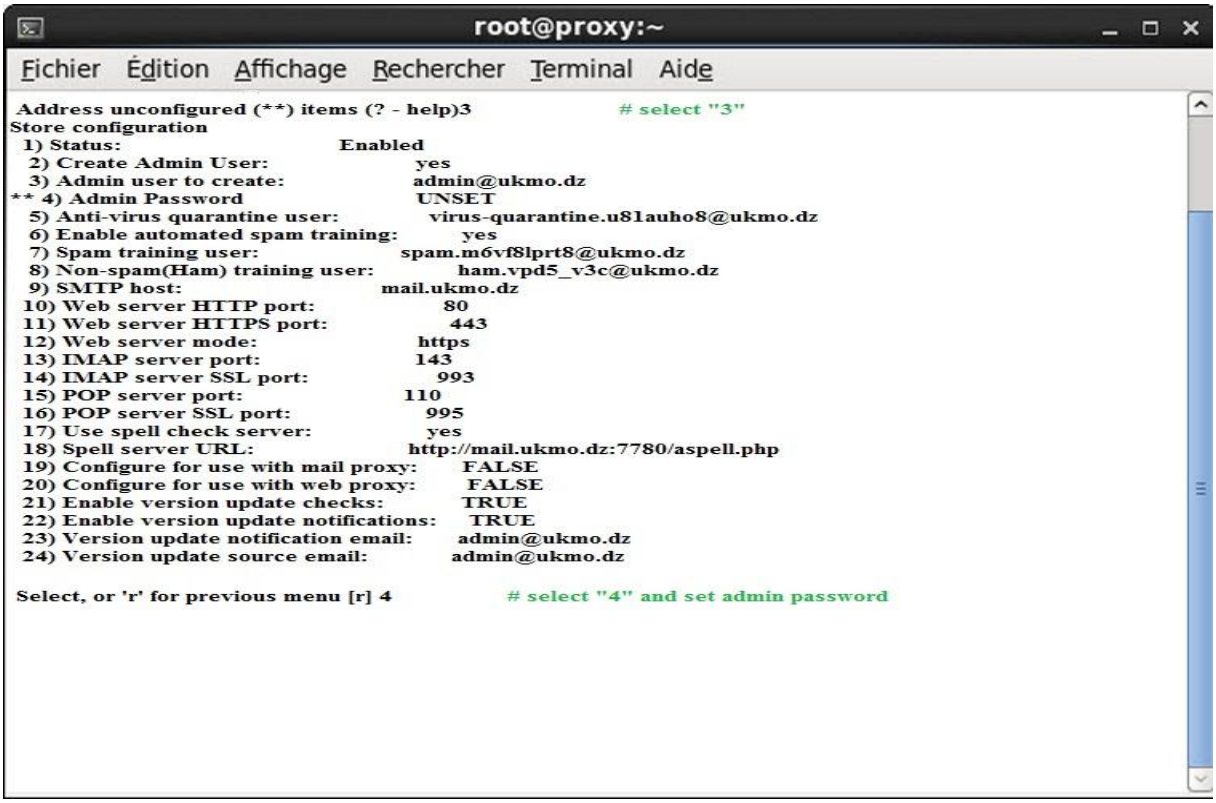


Figure 20: configurer le mot de passe d'administrateur

Mettez le numéro 3 de config Admin Password (numéro 4), puis mettre r de revenir et de mettre une configuration à appliquer Zimbra

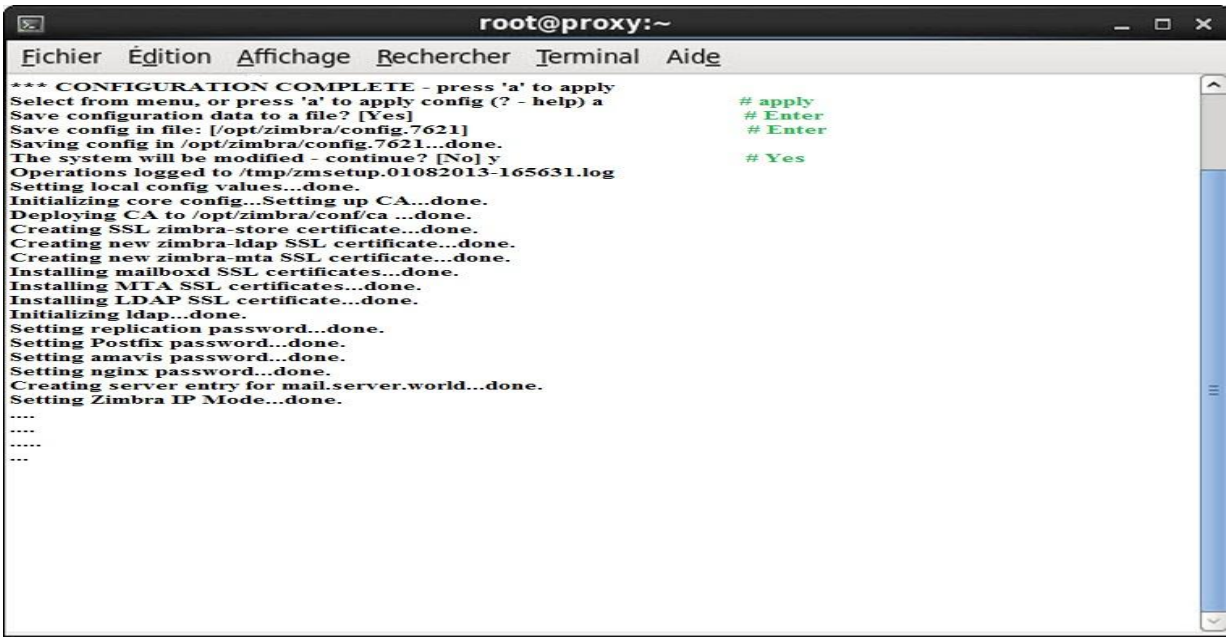


Figure 21: réglage de la création des paquets

L'enregistrement du configuration et le réglage de la création des paquets

Vous pouvez configurer les domaines DNS et plus tard avec la console d'administration

Vous voudrez peut-être donner un essai sur le bureau Zimbra comme un client qui prend en charge le travail hors connexion.

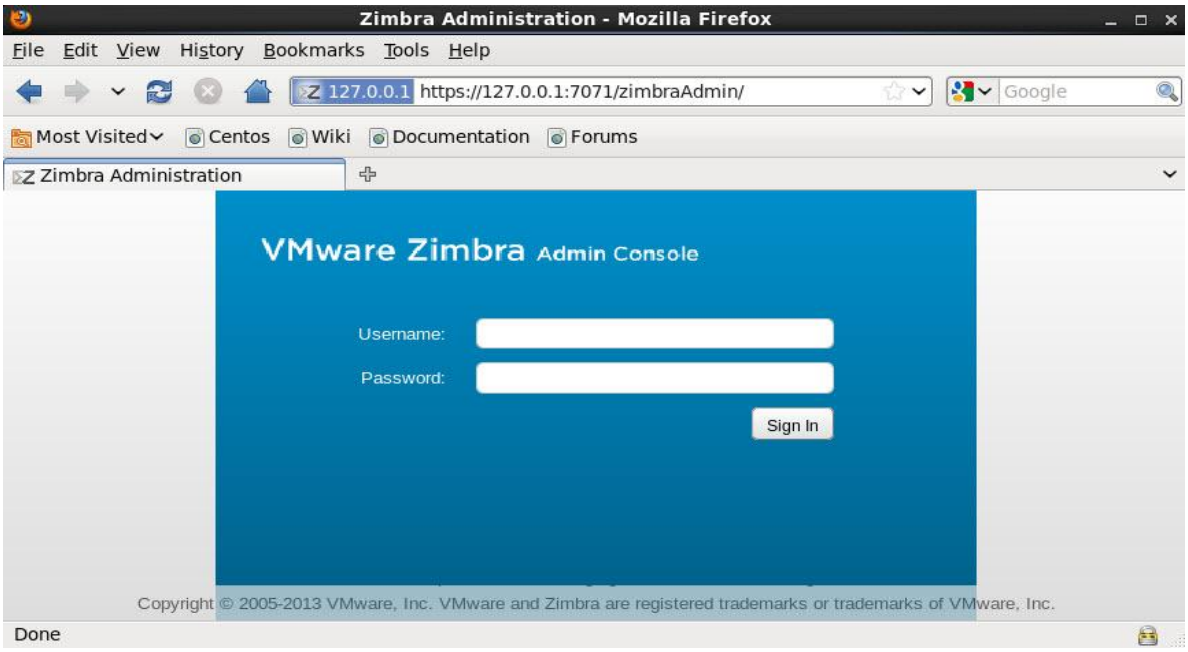


Figure 22: : l'interface de Zimbra

L'interface d'un accès au Zimbra

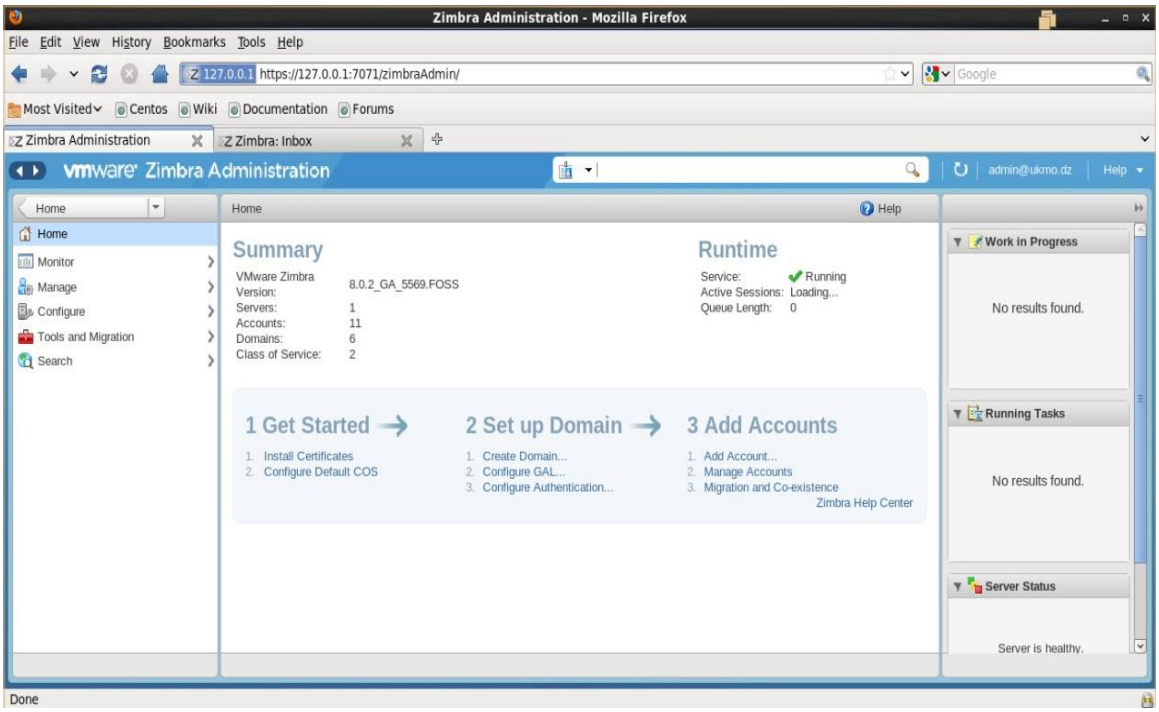


Figure 23: : l'interface d'un compte administrateur

L'interface d'un compte administrateur et les module qu'il peut untiliser pour leur configuration [14]

7. Installation Joomla

Dans une virtuel machine on intègre le système CentOS on essaye d'installer Joomla d'abord ont installer Wamp server et le Apache sur le serveur dont le conf.d qui a tous les besoin d'information du Apache alors on installe le httpd aussi les package de MySQL ensuite le l'installation du php .

Installation de joomla 3:

1. **–Installer Apache:**

   # yum groupinstall "Web Server"

2. **-Installer MySQL:**

   # `yum groupinstall "MySQL Database server"`

3. **– Télécharger et extraire Joomla :**
   - # cd /tmp/
   - # wget http://joomlacode.org/gf/download/frsrelease/16914/73507/Joomla3Stable-Full_Package.tar.gz
   - # mkdir /tmp/joomla3
   - # tar -zxvf Joomla_3-Stable-Full_Package.tar.gz -C /tmp/joomla/

4. **- Déplacez tous les fichiers vers le répertoire web à domicile:**

   # mv /tmp/joomla/* /var/www/html/

5. **Lancer MySQL et configuré pour démarrer au démarrage::**

   # `service mysqld start; chkconfig mysqld on`

6. **- Le mot de passe root pour MySQL et obtenir MySQL prêt à la production:**

   # `/usr/bin/mysql_secure_installation`

7. **Installer phpmyadmin du référentiel epel pour faciliter le maintien mysql**:

   # `yum --enablerepo=epel install phpmyadmin`

   - **Edit /etc/httpd/conf.d/phpMyAdmin.conf**

     # vi /etc/httpd/conf.d/phpMyAdmin.conf

     Note: line 14: add IP address you permit

     `Allow from 127.0.0.1 192.168.10.0/24`

Ouvrez le port http:

```
# iptables -I INPUT -p tcp --dport http -j ACCEPT ; service
iptables save ; service iptables restart
```

Tournez output buffering off en éditant/etc/php.ini change:

```
output_buffering=4096
```

to

```
output_buffering=Off
```

Créer un fichier vide configuration.php et définir des autorisations:

touch /var/www/html/configuration.php
chmod 666 /var/www/html/configuration.php

chown -R apache.apache /var/www/html/univ-ouargla

- Démarrer Apache et configuré pour démarrer au boot:

service httpd start; chkconfig httpd on

Cree la base de données:

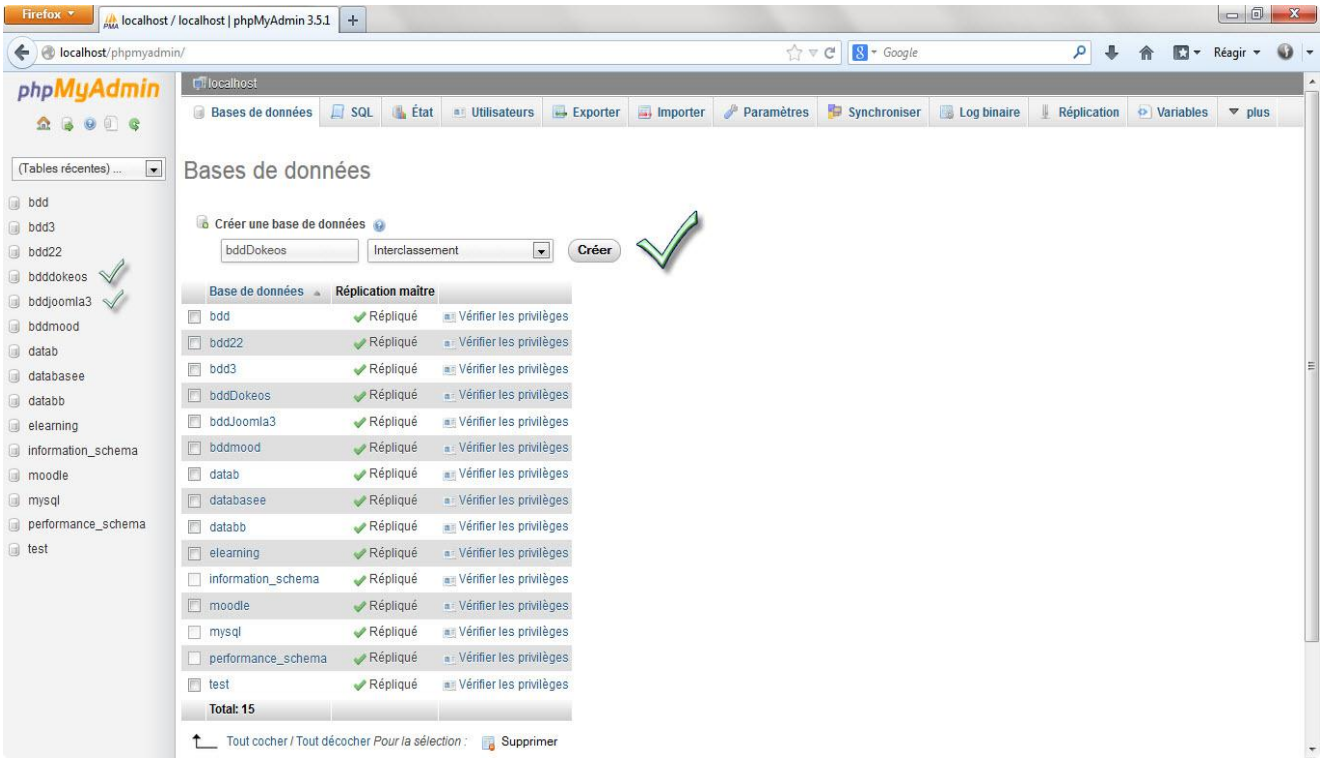


Figure 24: création de la base de donnée

Installer Joomla, ouverte le type de navigateur web: http://localhost/univ-ouargla

Avec les conditions ci-dessus remplies, une base de données créée et le Joomla requis fichiers en place, vous êtes prêt à installer Joomla. Démarrez le Joomla installateur web en naviguant sur le nom de domaine du site, www.ukmo.dz et vous devriez voir Joomla va tenter d'identifier le champ Select Language automatiquement à partir de la langue de votre navigateur. Vous pouvez modifier si nécessaire.

Remplissez les informations ci-dessous.

Nom du site: Le nom de votre site Web - ce qui peut être modifié à tout moment ultérieur dans la page de configuration globale du site.

Description: Entrez une description du site. Il s'agit d'une la meta description repli global utilisé sur chaque page qui sera utilisée par les moteurs de recherche

Administrateur Adresse e-mail: L'adresse électronique de l'administrateur.

Administrateur Nom d'utilisateur: Joomla utilise par défaut "admin" comme nom d'utilisateur pour le Super User.

Admin Mot de passe: entrez un mot de passe très fort.

Site hors ligne: Cliquez sur Oui ou Non boîte. Oui - cela signifie que l'installation est terminée, votre Joomla site Web affichera le. Non - ce qui signifie que le site est en direct lorsque vous accédez à yoursitename.com pour afficher la page d'accueil.

Quand tout sur la première page est terminé, cliquez sur le bouton Suivant pour

Figure 25: entrer les données principaux pour notre site

Nous avons besoin des informations sur la base de données qui a été suggéré ci-dessus à noter.

Pour simplifier, ces instructions sont une référence à l'installation d'une base de données MySQL. Les instructions sur la page d'installation sont explicites,

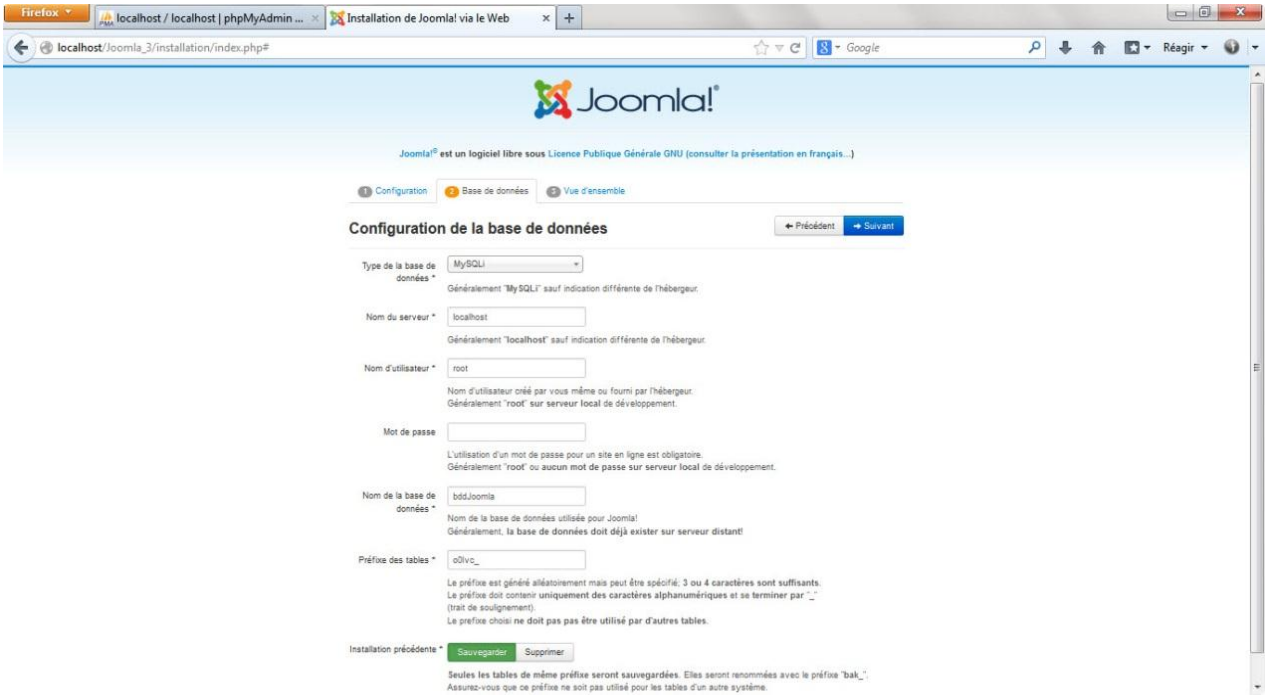


Figure 26: configuration de la base de donnée

Il est maintenant temps pour finaliser le Joomla installation. La dernière page de l'installation contient toutes les informations sur l'installation.

Si tout est en ordre, vous verrez l'installation en haut de la page de présentation. Sinon, c'est l'endroit pour vérifier et voir ce qui peut causer un problème.

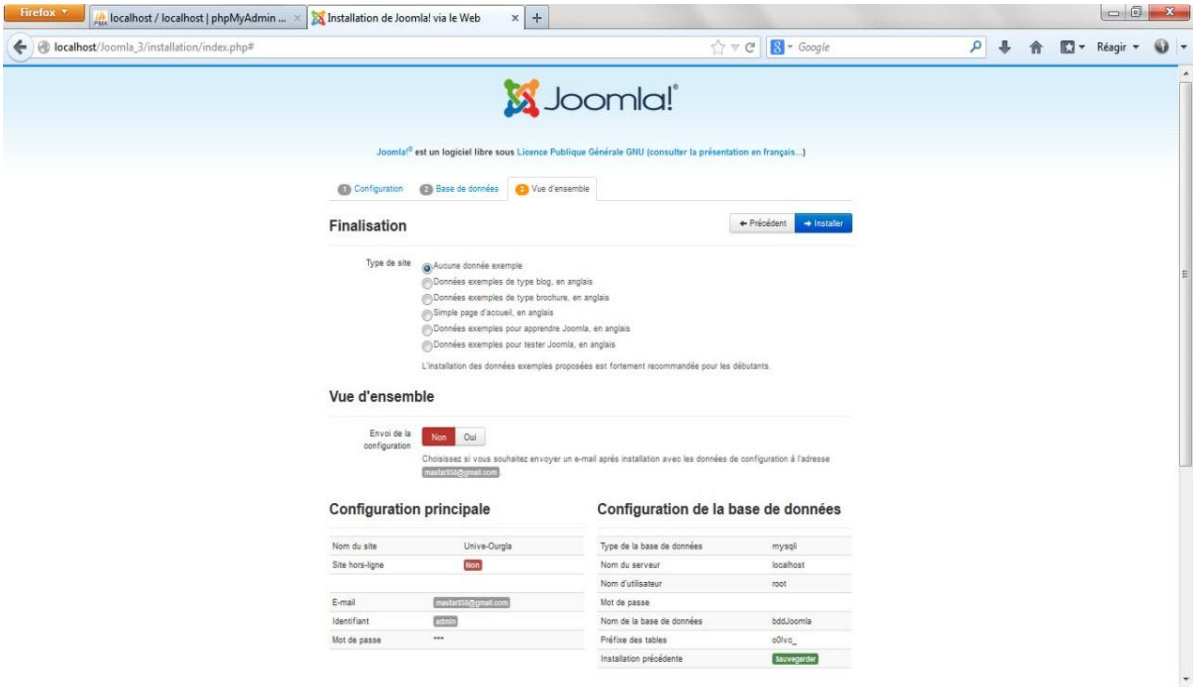


Figure 27 : la fin d'installation de Joomla

Figure 28 : Joomla en cour d'installation

Joomla 3 est maintenant installé, mais il ya une dernière étape pour terminer l'installation et commencer à utiliser votre site Joomla! Site propulsé. Vous devez supprimer le dossier d'installation. Cliquez sur le dossier d'installation Retirez et un message de réussite s'affiche. Maintenant, vous pouvez naviguer dans le journal de l'administrateur en cliquant administrateur ou aller droit vers votre site en cliquant sur le site.***[15]***

Figure 29: la dernière étape et suppression de dossier d'installation

Entrer comme un administrateur :http://localhost/joomla/administrator/

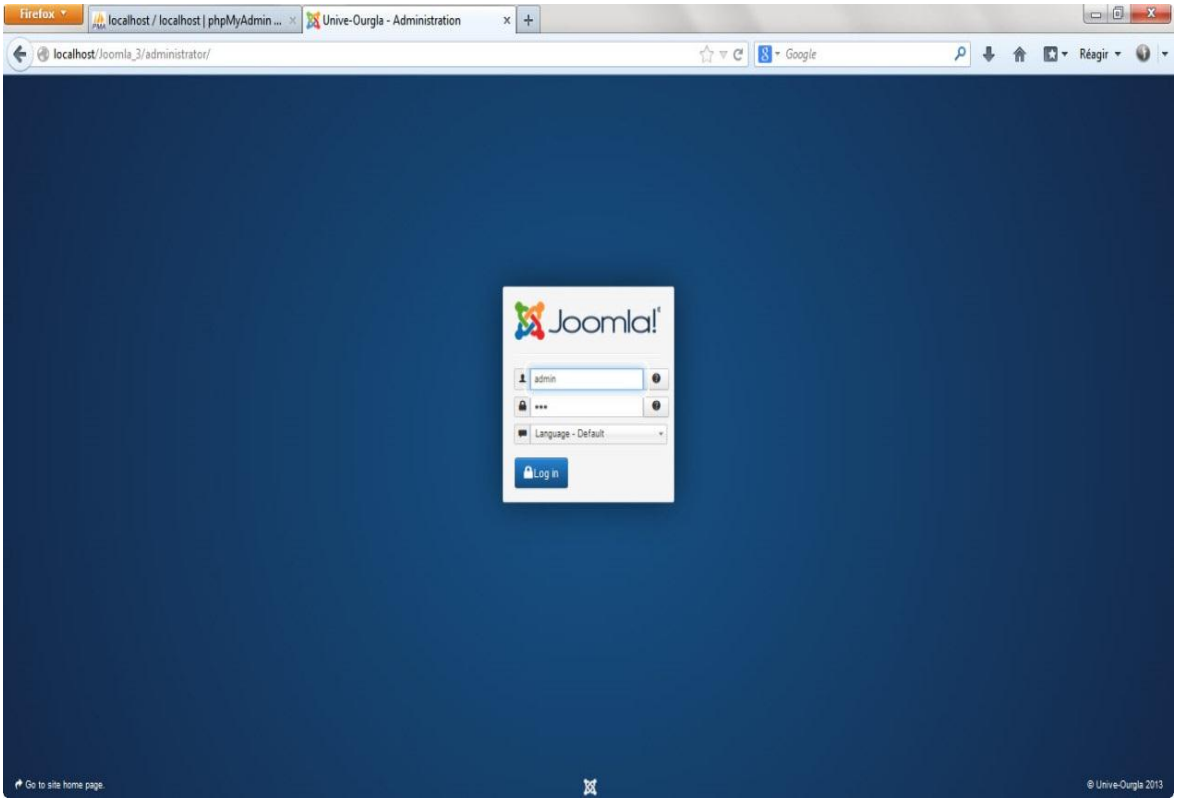


Figure 30: l'interface de joomla

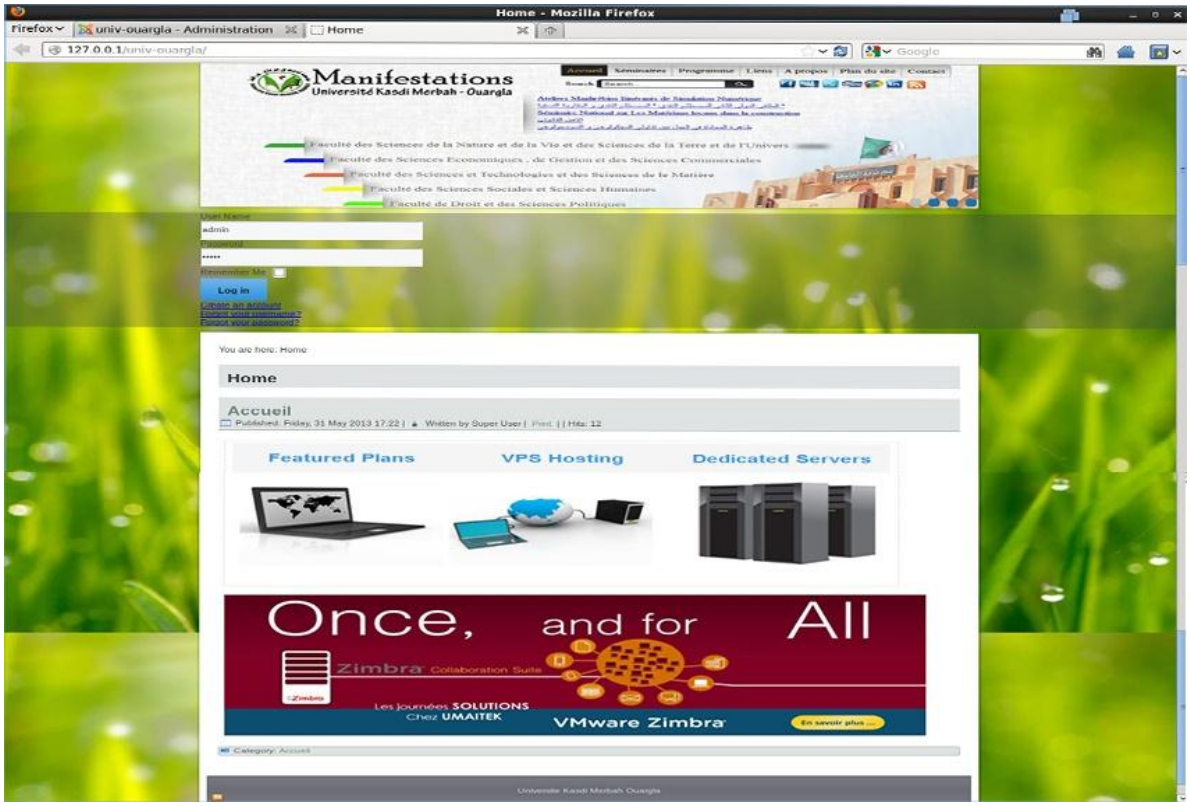


Figure 34: accéder a Joomla en tant que administrateur

8. L'installation de la plateforme DOKEOS

Installer Dokeos:

1. Télécharger et extraire Dokeos:

- # cd /tmp/
- # wget http://dokeos.org/gf/download/frsrelease/16914/73507/dokeos2.1Stable-Full_Package.tar.gz

2. Déplacez tous les fichiers vers le répertoire web à domicile:

mv /tmp/dokeos /var/www/html/

3.Créer vides Configuration.php autorisations de fichiers et ensemble:

touch /var/www/html/configuration.php

chmod 666 /var/www/html/configuration.php

cd dokeos

chmod –R 0777 /main/install/ main/inc/conf/ main/upload/template_thumbanils/

main/upload/users/ main/default_course_document/images/ archive/coures/ home/

#chown -R apache.apache /var/www/html/univ-ouargla

*L'accès à la plateforme Dokeos par l'exécution de cette instruction dans la barre des

adresses au niveau du navigateur : http://localhost/dokeos/

Figure 31: le début d'installation de Dokeos

tape 1 : Suivre les instructions (choix de la langue)

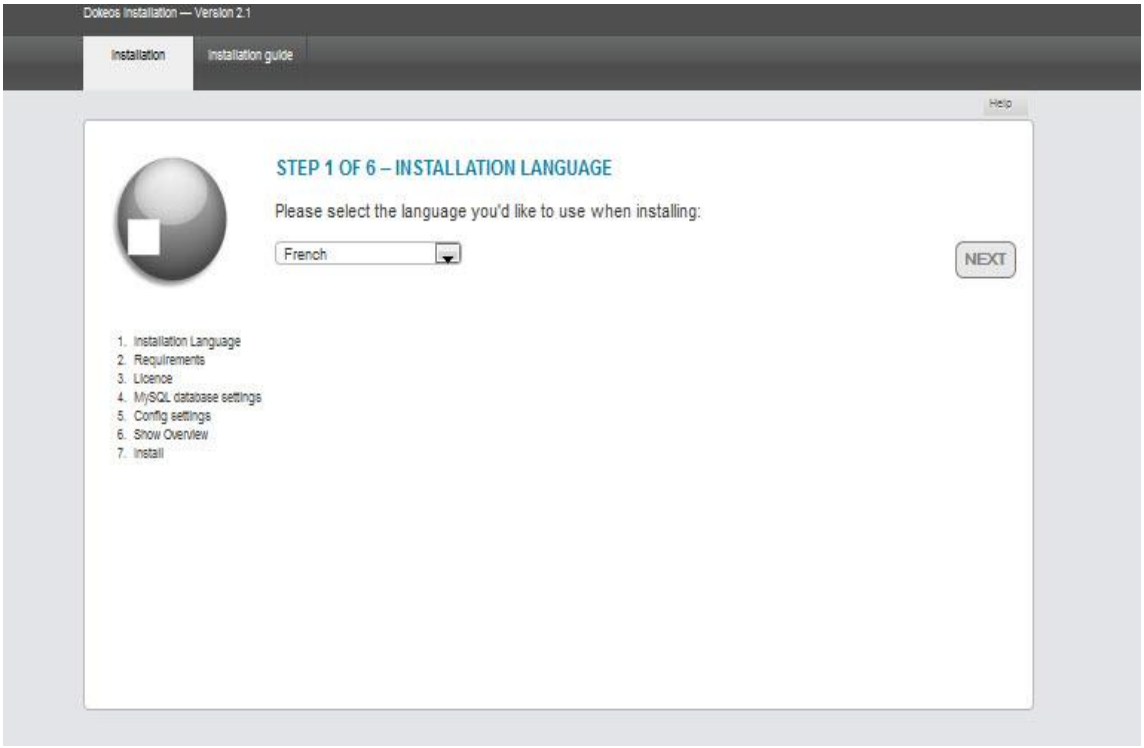


Figure 32 :installation de language

Etape2 :dans cette étape on voir les fonctionnalités que le serveur doit être en mesure pour utiliser dokeos et le paramètre qui sont attribuer dans le fichier de configuration.php.ini sur le serveur et les fichiers qu'on peut les modifier et les écrit dans le serveur web c'est une modification manuel .

Figure 33: les fonctionnalistes de Dokeos

Etape3 : ici la licence de DOKEOS

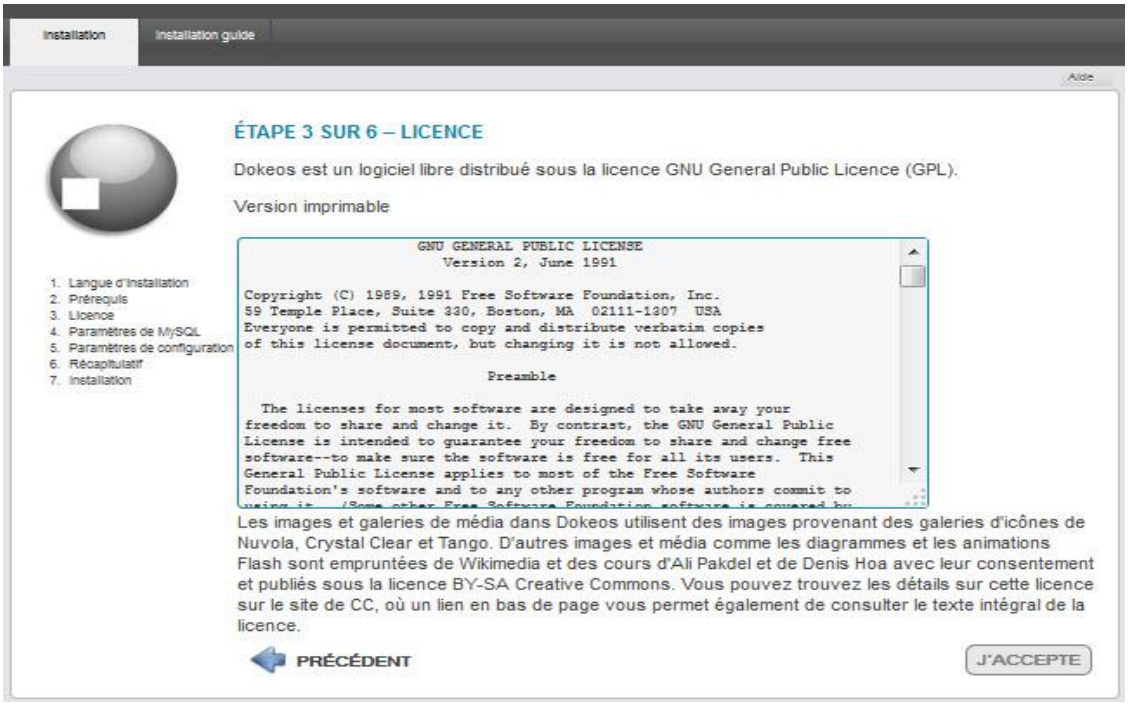


Figure 34: la licence de Dokeos

Etape 4 : paramètre MySQL pour les données de mots de passe

Figure 35: paramètre de MySQL

Etape5 : ici les paramètres de configuration ,L'essentiel de nos données personnelles d'administration du site

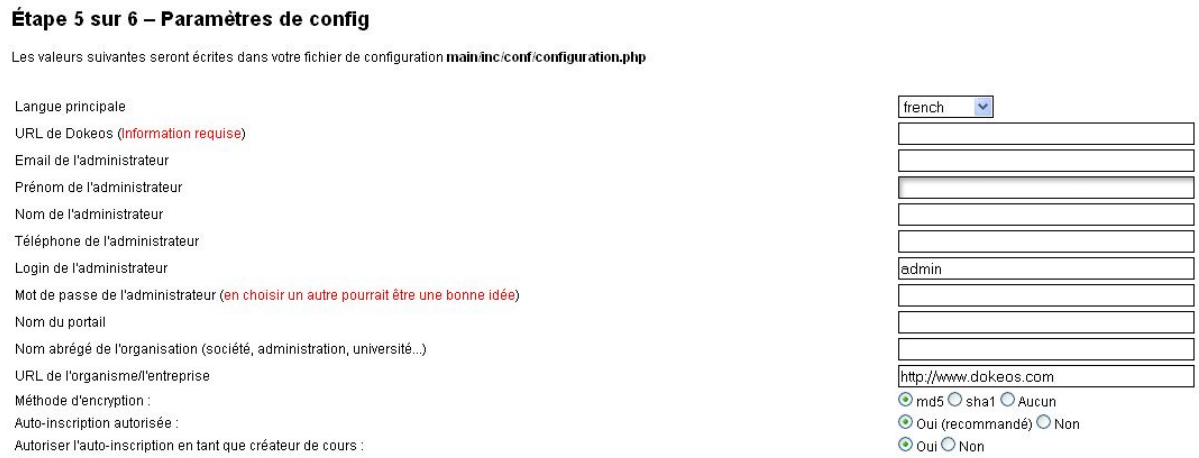


Figure 36: paramètre de configuration

Etape6 : Contrôle des données, Rappel pour vérification des Paramètres de configuration

Figure 37 : dernière vérification avant installation

Et voilà la page d'accueil de Dokeos ***[16]***

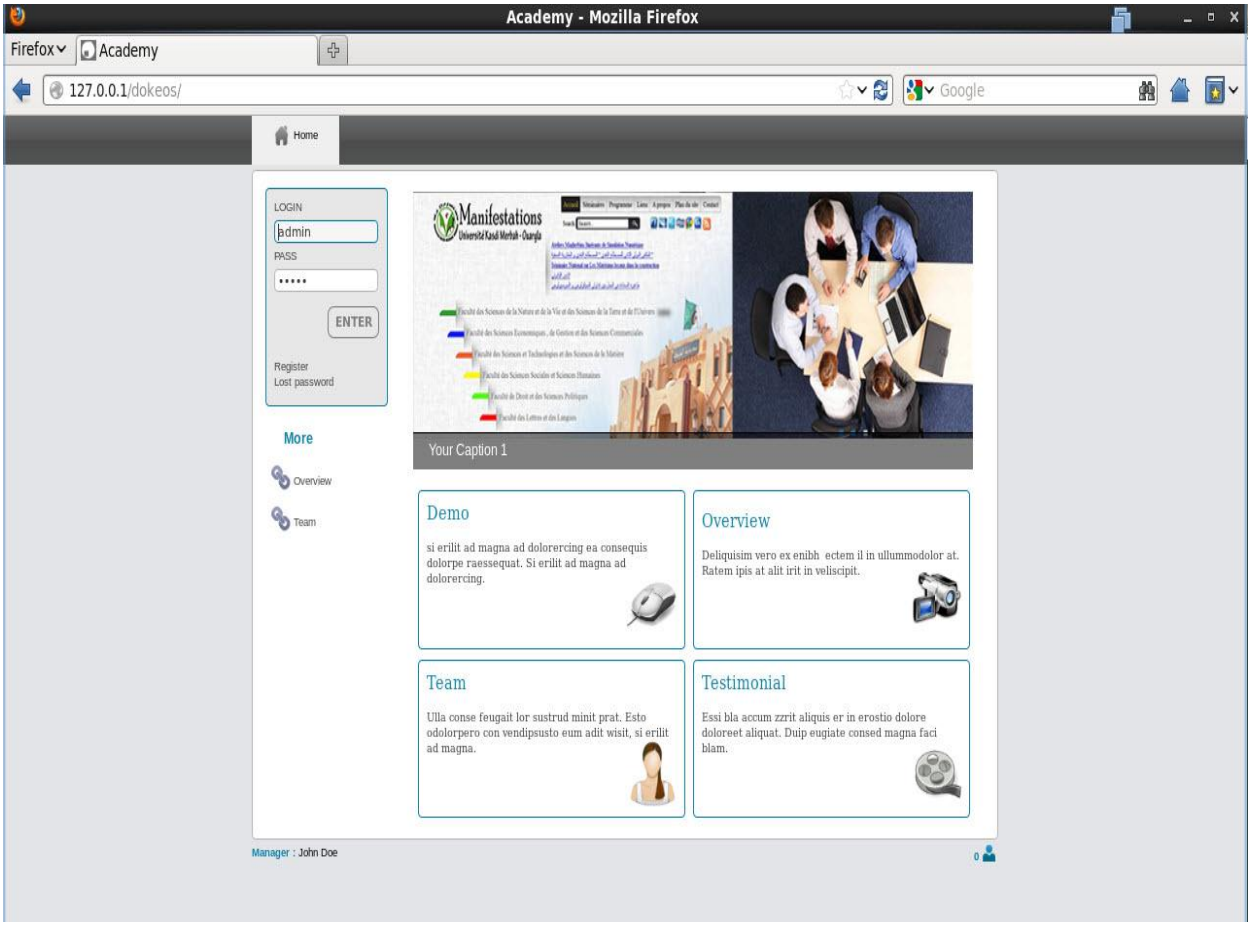


Figure 38: page d'accueil de Dokeos

9. Les méthodes utilisé dans notre travail

8.1. La première méthode

Dans cette méthode on a installé Open LDAP (serveur) comme un annuaire et on a configuré Zimbra pour cet Open LDAP, tous les comptes des utilisateurs sont regroupé dans cette annuaire ils vont accéder aux autre plateformes Zimbra, Joomla et Dokeos avec un couple de (identifiants, mots de passe) unique comme c'était présenter sur cette figure ci-dessous.

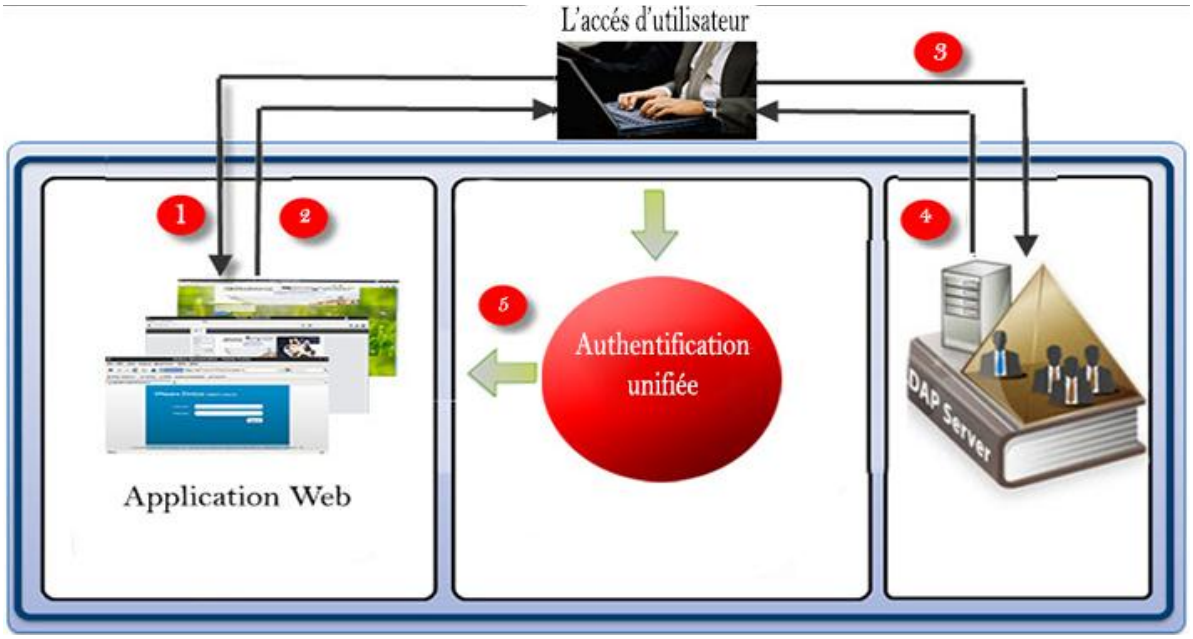


Figure 39: Authentification unifiée avec un Open LDAP

1) l'utilisateur doit demander l'interface d'une plateforme.
2) l'interface apparait.
3) l'utilisateur dois saisir son identifiant et mot de passe, LDAP doit vérifier cette utilisateur s'il est intégré au LDAP ?
4) LDAP doit répondre à cette utilisateur est qu'il l'autorise l'accès aux plateformes ou non.
5) Si cet utilisateur est accepté, il peut accéder aux plateformes.

C'est la méthode la plus connue et utilisée dans les entreprises.

8.2. La deuxième méthode :

Dans cette méthode on a installé Ldap Zimbra qui est un LDAP avec le serveur de messagerie Zimbra.

Ce ldap zimbra est configurer avec Zimbra lui-même, alors au lieu d'installer un ldap ensuite le configurer, on va réduire le travail et on utiliser ldap zimbra et on termine la configuration de Dokeos et Joomla, cette méthode est plus facile que la première car on fait l'ajout directement au zimbra.

Figure 40: Authentification unifiée avec LDAP Zimbra

1) L'utilisateur doit accéder à la plateforme Zimbra, on ajoute les utilisateurs au zimbra direct qui intègre un ldap interne, l'administrateur crée et ajoutera des utilisateurs directs au Zimbra.
2) LDAP Zimbra sera comme un annuaire pour se connecter et configurer avec les autre plateformes, l'accès sera avec le même (identifiant, mot de passe) pour les trois : Zimbra et Dokeos (LMS) , Joomla (CMS).

Conclusion général

L'authentification multi plateformes est un aspect fédérateur pour les applications du système informatique. C´est avant tout un système d´administration. En termes de sécurité, le code d´accès unique pour toutes les plates-formes peut fragiliser les accès aux applications (un mot de passe donne accès à l´ensemble des applications de l´utilisateur). Un effort de sécurisation du réseau doit donc être fait en parallèle et être concentré au niveau du serveur d´authentification. Un système d´authentification unique doit s´intégrer dans une politique de sécurité.

La mise en place de notre travail qui concerne l'authentification multiplateforme peut simplifier l'accès à différentes plateformes de l'environnement universitaire avec une seule connexion, une confidentialité et bonne sécurité pour l'utilisateur. Ce travail est basé sur un serveur LDAP qui est un annuaire pour notre base de données qui contient les enseignants et les étudiants. L´annuaire LDAP permet un gain d´efficacité important sur les tâches d´administration.

Pour notre travail nous avons conclue qu'il est valable pour l'environnement universitaire car on a choisi les plateformes qui sont utiles à l'université : Dokeos comme un e-learning, Zimbra comme un serveur de messagerie et Joomla est valable pour faire un site d'université complet et présentable.

Bibliographie :

[01]:CALINE VILLACRES & ERNST& YOUNGLLP, *L'Authentification de A à Z* .

[02] : *Comparatif de VMware Zimbra aux principales plates-formes de messagerie et de collaboration* , Copyright © 2011 VMware

[03] : *Construire et animer des formations en ligne avec Dokeos 2.0*,© Dokeos , Mars 2011

[04] : GANAEL LAPLANCHE , *Formation Open LDAP* ,école ouverte francophone , 2005-2010.

[05] : HAGEN GRAF &JEN KRAMER & MILENA MITOVA & ANGIE RADTKE , *Joomla! 2.5 Le Guide Pour Débutant.*

[06] : IVAN GAUTREAUALIAS HORNOS , *Joomla 1,5 ! pour les nuls* , novembre 2010

[07] : OLIVER SALAUM ,*Introduction aux architectures web de Single Sign-on***,**15 Octobre 2003.

[08] PALO ALTO , *VMware Zimbra Collaboration Server Administrator's Guide ZCS 8.0* , California 94304 USA, August 2012 .

[09] :STEPHANE VINSOT Vinsot , *Les 7 méthodes d'authentification les plus utilisées*, Evidian , 2007

[10] :VINCENT MATHIEU & PASCAL AUBRY & JULIEN MARCHAL , *Single Sign-On open-source avec CAS (Central Authentication Service)* , 2003.

Webographie

[11] :http://www.centosadmin.net/pourquoi-centos.htmlle site officiel de CentOS consulter le 05/04/2013.

[12] : http://www.server-world.info/en/note?os=CentOS_6&p=dns&f=1 consulter le 25/04/2013 .

[13] : http://www.server-world.info/en/note?os=CentOS_6&p=ldap consulter le 04/05/2013.

[14]: http://www.silverlake.fr/index.php?post/2011/11/24/Zimbra-7-sur-Centos-6-64-bits consulter le 29/05/2013.

[15] : http://www.support-joomla.com/installer-joomla/installation-de-joomla consulter le 01/06/2013.

[16]: http://www.dokeos.com/fr/node/10779 consulter le 01/06/2013.

[17]: http://www.mindflash.com/learning-management-systems/what-is-lms consulter le 25/05/2013.

[18] : http://www.aepik.net/documentation/securite/sso consulter le 10/06/2013.

[19] :http://www.esup-portail.org/consortium/espace/SSO_1B/cas/ consulter le 10/06/2013.

[20] http://www.krizna.com/centos/how-to-install-dns-server-in-centos-6/ consulter le 01/04/2013.