République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master Professionnel

Option : Administration et Sécurité des Réseaux

Thème

La mise en œuvre d'un système d'authentification unique SSO avec CAS.

Réalisé par :

M^{lle} BOUKOUCHA Hana

M^{lle} BOUKTIT Nadjet

Devant le jury composé de :

Président : M^{r} Z.FARAH

Examinateur : M^{r} R.OUZEGGANE

Encadreur : M^{me} M.YAICI

Promotion 2021 - 2022

Remerciements

Nos vifs remerciements en premier lieu à Dieu tout puissant qui nous a doté d'une grande volonté, patience et force pour mener à terme notre projet.

Notre profonde gratitude et sincères remerciements à notre encadrante Mme M.Yaici qui a toujours fait preuve de disponibilité et qui nous a éclairés tout au long de notre travail par ses conseils et orientations quant à l'élaboration de ce projet.

Nous tenons à exprimer toute notre grande gratitude aux membres de jury d'avoir accepté de juger ce travail.

Nous remercions nos chers parents pour leur soutien, confiance et leurs précieux conseils qui nous ont éclairés dans l'accomplissement de ce travail.

Nos très vifs remerciements à tous ceux qui ont contribué de près ou de loin et en particulier l'ensemble des enseignants du département d'Informatique de l'Université **ABDERRAHMANE MIRA** de **BEJAIA**.

Dédicaces

A celle qui a attendu avec patience les fruits de sa bonne éducation et de ses dévouements

A ma chère mère ***Nouara***

A celui qui s'est changé la nuit en jour pour m'assurer les bonnes conditions

A mon cher père ***Abdellah***

A mes frères **Lounis** et **Oualid**, pour leur appui et leurs encouragements

A mes chères sœurs Ghania, Fahima, Fatiha, Nachida, Hada et mes belles sœurs Hassiba et Soulef, pour leurs présence à mes côtés et leurs soutiens.

A mes chers neveux et nièces ***Rahim***, ***Malak***, ***Safa***, ***Ayane***, ***Arris***, ***Rital***, ***Anes***, ***Chaima***, ***Aksil*** *et* ***Soltana***

A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité

A mon chers binôme ***Nadjet***

A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de succès

A tous ceux que j'aime

Je dédie ce modeste travail

Hana

Dédicaces

Je dédie ce travail, à mon papa adoré et à la prunelle de mes yeux, ma maman chérie,

C'est grâce à votre affection, amour, soutien et confiance en moi qui m'ont permis d'arriver là où je suis. Que ce travail soit la reconnaissance à tous vos sacrifices.

A mes chères sœurs et frères Nabila, Hakima, Ahmed et à mon beau-frère Riad et ma princesse Kayla,

Merci pour tout ce que vous m'avez apporté, joie, bonheur, encouragements, affection et aide.

A mon petit Youcef et ma très chère grand-mère.

A la mémoire de ma grand-mère et mon grand-père,

Que dieu vous accueil dans son vaste paradis.

A mes amies Lynda, Rosa, Katia, Taous, Hanane, Melissa, Samia, Sabrina, Dyhia,

Merci pour votre présence et pour votre soutien.

A ma chère binôme Hana,

A une personne particulière, Yacine,

Tes encouragements et ta présence ont étés mon appui.
Merci pour tout ce que tu m'as apporté.

A toutes les personnes qui m'ont apporté de l'aide et contribué à la réalisation de ce projet.

Nadjet

Table des matières

Table des matières i

Table des figures iii

Liste des tableaux iv

Acronymes v

Introduction générale 1

I Généralités sur la sécurité informatique et l'authentification 2
- I.1 Introduction . . . 2
- I.2 **La sécurité informatique** . . . 2
  - I.2.1 **Terminologie de la sécurité informatique** . . . 3
  - I.2.2 **Types de menaces** . . . 3
  - I.2.3 **Types d'attaques** . . . 4
  - I.2.4 **Les mécanismes de sécurité** . . . 6
- I.3 **l'authentification** . . . 8
  - I.3.1 **Définition** . . . 8
  - I.3.2 **Facteurs d'authentification** . . . 8
  - I.3.3 **Méthodes et types d'authentification** . . . 9
- I.4 **Protocoles d'authentification** . . . 10
- I.5 **Conclusion** . . . 10

II Le Single Sign-On et Le mécanisme CAS 11
- II.1 Introduction . . . 11
- II.2 **Définition** . . . 11
- II.3 **Objectif du Single Sign-On** . . . 12
- II.4 **Avantages et inconvénients du Single Sign-On** . . . 13
- II.5 **Architecture d'un SSO** . . . 13
  - II.5.1 **SSO côté client :** . . . 14

II.5.2 **SSO côté serveur :** . . . 15
II.5.3 **SSO hybrides :** . . . 17
II.6 **Les différentes approches du SSO** . . . 18
II.6.1 **L'approche centralisée :** . . . 18
II.6.2 **L'approche fédérative :** . . . 18
II.6.3 **L'approche coopérative :** . . . 20
II.7 **Les solutions d'un SSO existantes** . . . 20
II.8 **CAS (Central Authentification Service) :** . . . 21
II.8.1 **Définition** . . . 21
II.8.2 **Mécanisme de CAS** . . . 21
II.9 **Authentification avec l'annuaire LDAP** . . . 26
II.9.1 **Les annuaires LDAP :** . . . 26
II.9.2 **Utilité de LDAP :** . . . 26
II.10 **Conclusion** . . . 27

III Mise en œuvre **28**
III.1 Introduction . . . 28
III.2 **Présentation du projet** . . . 28
III.3 **Installation et configuration des serveurs** . . . 29
III.3.1 **OpenLDAP** . . . 29
III.3.2 **Installation du serveur d'authentification unique Apereo-Central Authentification Server (CAS) 6** . . . 34
III.4 **Réalisation de l'interface d'authentification avec le service Java-RMI méthodes** . . . 37
III.5 **Conclusion** . . . 38

Conclusion générale **39**

Annexes **41**

Bibliographie **43**

Table des figures

I.1 Attaque par rebond. 5
I.2 Attaque indirecte par réponse. 6
I.3 Chiffrement symétrique. 7
I.4 Chiffrement asymétrique. 7
I.5 Méthodes d'authentification. 9

II.1 Principe du SSO. 12
II.2 SSO coté client avec serveur d'informations d'identification. 15
II.3 SSO coté client avec un agent de serveur d'authentification. 16
II.4 SSO coté client serveur avec agent de reverse proxy. 17
II.5 SSO centralisé. 18
II.6 La fédération du SSO. 19
II.7 Premier accès d'un navigateur au serveur CAS. 22
II.8 Authentification d'un navigateur auprès du serveur CAS. 22
II.9 Redirection par le serveur CAS d'un navigateur vers un client CAS après authentification. 23
II.10 Récupération d'un PGT par un mandataire CAS auprès du serveur CAS [1] 24
II.11 Validation d'un PT par un client CAS accédé par un mandataire CAS [31]. 25
II.12 Fonctionnement de n-tiers. 25
II.13 LDAP au cœur d'un système. 27

III.1 L'architecture du système. 29
III.2 Configuration du nom de domaine. 30
III.3 Teste de la configuration du DNS. 31
III.4 Résultat de la configuration de LDAP. 32
III.5 Interface Web de LDAP. 33
III.6 Interface d'authentification de CAS. 37
III.7 Interface de l'application. 38

Liste des tableaux

II.1 Solution du SSO [2] 20

Listes des acronymes

3DES	3 Data Encryption Standard
AES	Advanced Encryption System
CAS	Central Authentification Service
CA	Certification Authority
CHAP	Challenge Handshake Authentification Protocol
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
HTTPS	Hypertext Transfer Protocol secure
IDP	Identifier Provider
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
NIS	Network Information Service
PAP	Password Authentification Protocol
PGT	Proxy Granting Ticket
PT	Proxy Ticket
RADIUS	Remote Access Dial In User Service
RC4	Rivest Cipher 4
RSA	Rivest, Shamir et Adleman
SI	Système d'Information
SP	Service Provider
SQL	Structured Query Language

SSH	Secure Shell
SSL/TLS	Secure Socket Layer/Transport Layer Security
SSO	Single Sign-On
TCP	Transfer Control Protocol
TGC	Ticket Granting Cookie
TS	Ticket Service
URL	Uniform Resource Locator

Introduction générale

Dans le monde numérique, les utilisateurs ont accès à plusieurs systèmes (service Web, Application ... etc). À mesure que le nombre d'informations d'identification pour chaque utilisateur augmente, la possibilité de les perdre ou de les oublier augmente également.

Le Single Sign-On ou l'authentification unique réduit le risque pour les administrateurs de gérer les utilisateurs de manière centralisée, augmente la productivité des utilisateurs en permettant la mobilité et permet aux utilisateurs d'accéder à plusieurs services ou applications après s'être authentifié une seule fois. Le SSO apporte un confort non négligeable à l'utilisateur et permet au système informatique de tracer facilement ces accès.

Notre travail consiste à réaliser un système d'authentification unique SSO Pour cela, nous avons à notre disposition plusieurs solutions d'authentification toutes basées sur un même mécanisme d'authentification unique. Ce dernier nécessite la combinaison de plusieurs composants, notamment un annuaire (LDAP) pour contenir l'ensemble des utilisateurs et unifier l'authentification des applications et un serveur d'authentification unique pour synchroniser l'accès aux applications (CAS).

Ce mémoire est organisé comme suit :

Chapitre I : Dans ce chapitre nous avons représenté les déférents concepts de sécurité et l'aspect de l'authentification qui est important dans notre application web, nous avons montré aussi leurs techniques, les avantages et les inconvénients et les différentes méthodes d'authentification.

Chapitre II : Dans ce chapitre nous avons abordé les techniques d'authentification, le Single Sign-On et le Central Authentication Service.

Chapitre III : Mise en œuvre, qui présente les diverses étapes ainsi que les différents outils adopté à la réalisation de notre projet.

Nous terminerons par une conclusion générale en décrivant les éléments essentiels qui ont été développés dans ce mémoire.

Chapitre I

Généralités sur la sécurité informatique et l'authentification

I.1 Introduction

La sécurité des systèmes informatiques se limite généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés. Dans ce chapitre, nous allons aborder les différents aspects liés à la sécurité informatique ainsi que les différentes notions de l'authentification.

I.2 La sécurité informatique

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles. L'objectif de la sécurité informatique est d'assurer que les ressources matérielles et/ou logicielles d'un parc informatique sont uniquement utilisées dans le cadre prévue par les personnes autorisées [3].
Nous allons commencer par identifier les exigences fondamentales en sécurité informatique qui caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques au regard de la sécurité :

- **La confidentialité :** l'information ne doit pas être divulguée à toute personne, entité ou processus non autorisé. En clair, cela signifie que l'information n'est consultable que par ceux qui ont d'y accéder [4] ;
- **L'intégrité :** il faut garantir à chaque instant que les données qui circulent sont bien celles que l'on croit, et qu'il n'y a pas eu d'altération (volontaire ou non) au cours de la communication. L'intégrité des données doit valider l'intégralité des données, leur précision, leur authenticité et leur validité [3] ;
- **La disponibilité :** l'information doit être rendue accessible et utilisable sur demande par une entité autorisée. Cela veut dire que l'information doit être disponible dans des conditions convenues à l'avance [4] ;

- **La non-répudiation :** une transaction ne peut être niée par aucun des correspondants. La non-répudiation de l'origine et de la réception des données prouve que ces dernières ont bien été reçues. Cela se fait par le biais de certificats numériques grâce à une clé privée [3] ;
- **L'authentification :** elle limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données [3].

I.2.1 Terminologie de la sécurité informatique

Parmi les mots-clés de la sécurité qui sont largement repris dans la littérature informatique, nous distinguons les suivants :

- **Les vulnérabilités :** ce sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vu dans sa globalité présente des vulnérabilités qui peuvent être exploitables ou non [5]. Une erreur de configuration d'un équipement réseau constitue une vulnérabilité tout comme un mot de passe vide ou trivial ;
- **Les attaques (exploits) :** elles représentent les moyens d'exploitation d'une vulnérabilité. Il peut y avoir plusieurs attaques pour une même vulnérabilité, mais toutes les vulnérabilités ne sont pas exploitables [5] ;
- **Les contre-mesures :** ce sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique [5] ;
- **Le mécanisme de sécurité :** un mécanisme qui est conçu pour détecter, prévenir et lutter contre une attaque de sécurité [6] ;
- **Le service de sécurité :** un service qui augmente la sécurité des traitements et des échanges de données d'un système. Un service de sécurité utilise un ou plusieurs mécanismes de sécurité [6] ;
- **Le risque :** c'est la probabilité qu'une menace exploite une vulnérabilité. Autrement dit, c'est une possibilité qu'un fait dommageable se produise.

I.2.2 Types de menaces

Les menaces peuvent être vues comme des violations potentielles de la sécurité qui existent en raison des vulnérabilités du système. Nous distinguons deux types de menaces :

- **Menaces accidentelles :** ce sont celles qui existent sans qu'il y ait préméditation. Nous citons comme exemples de menaces accidentelles la défaillance de système, les bugs dans les logiciels et les bévues opérationnelles. [7]
- **Menaces intentionnelles :** c'est l'ensemble des actions malveillantes qui constituent la plus grosse partie du risque. Elles font l'objet principal des mesures de protection. Parmi elles, il y'en a deux catégories d'attaques :

1. **Attaques actives :** Ce sont les attaques dans lesquelles l'attaquant tente de modifier l'information ou de créer un faux message. Les attaques actives sont sous forme d'interruption, de modification et de fabrication. La prévention de ces attaques est assez difficile en raison d'un large éventail de vulnérabilité physique, de réseaux et de logiciels [8].
2. **Attaques passives :** Ce sont les attaques dans lesquelles l'attaquant se met en écoute non autorisée en surveillant simplement la transmission ou la collecte d'informations. Cette catégorie d'attaque n'apporte aucun changement aux données ou au système [8].

I.2.3 Types d'attaques

Les personnes malveillantes utilisent plusieurs techniques d'attaques qui peuvent être regroupées en trois catégories différentes :

I.2.3.1 Les attaques directes

Ce sont les plus simples des attaques. Le hacker attaque directement sa victime à partir de son ordinateur. En effet, les programmes de hacking qu'ils utilisent ne sont que faiblement paramétrables, et un grand nombre de ces logiciels envoient directement les paquets à la victime [9].

I.2.3.2 Les attaques indirectes par rebond

Cette attaque est très prisée des hackers, car le rebond a deux avantages :
— Masquer l'identité (l'adresse IP) du hacker ;
— Éventuellement, utiliser les ressources de l'ordinateur intermédiaire parce qu'il est plus puissant (CPU, bande passante...) pour attaquer.

Les paquets d'attaques sont envoyés à l'ordinateur intermédiaire qui répercute l'attaque vers la victime, d'où le terme de rebond [9]. (Figure I.1).

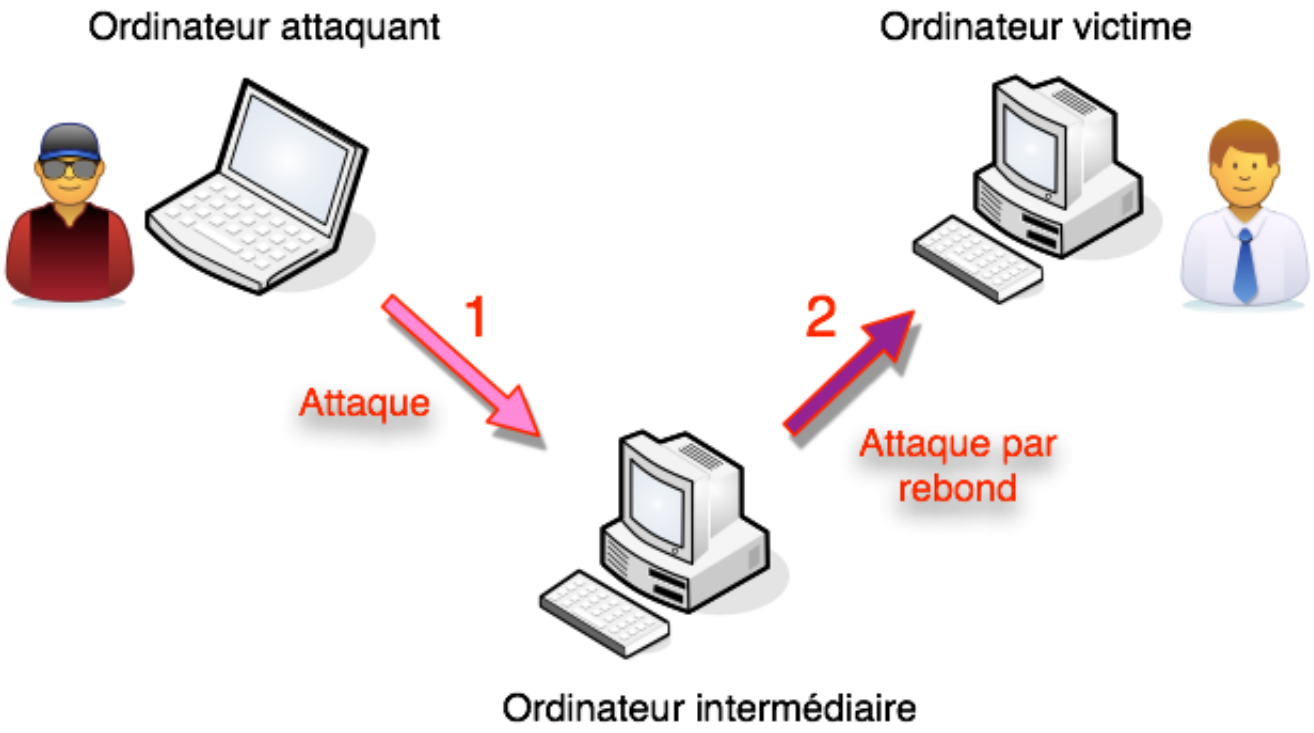

Figure I.1 – Attaque par rebond.

I.2.3.3 Les attaques indirectes par réponse

Cette attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages du point de vue du hacker. Mais, au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête. Et c'est cette réponse à la requête qui va être envoyée à l'ordinatrice victime [9]. (Figure I.2) :

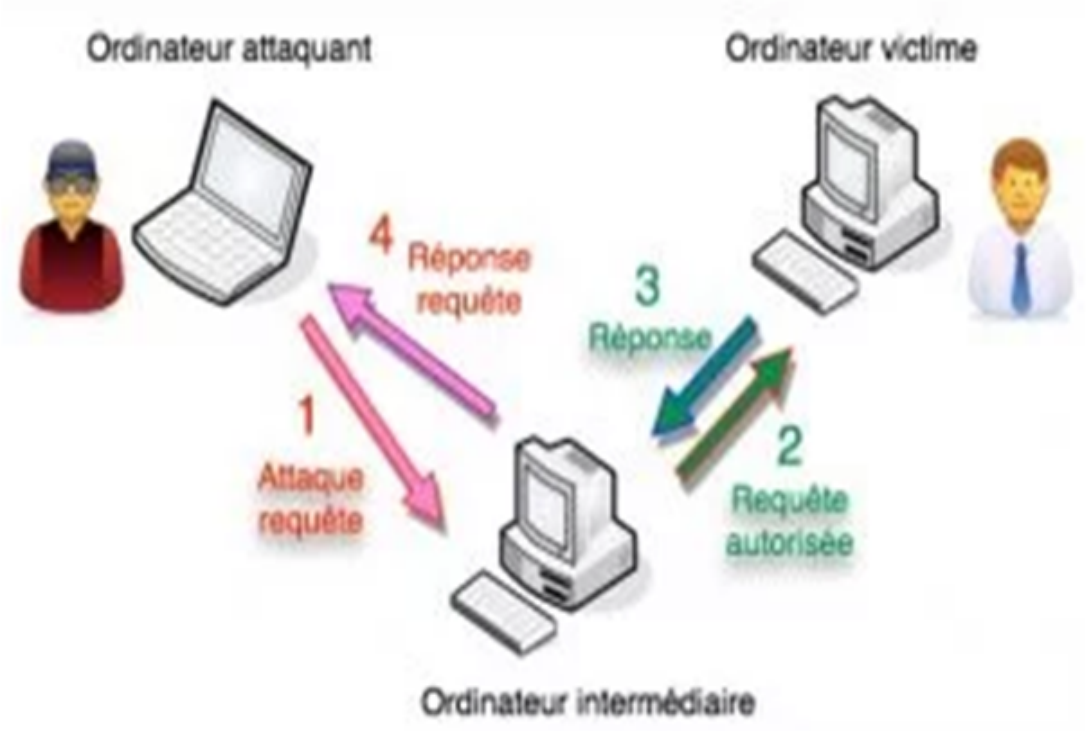

Figure I.2 – Attaque indirecte par réponse.

I.2.4 Les mécanismes de sécurité

Il existe deux grandes techniques de cryptographie : la cryptographie symétrique et la cryptographie asymétrique.

I.2.4.1 La cryptographie

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, ainsi les rendre simplement inintelligibles sans une action spécifique. Pour assurer les objectifs de la cryptographie, nous pouvons utiliser des algorithmes basés sur des clés.

Il existe deux types d'algorithmes de chiffrement :

— **Chiffrement symétrique (à clé secrète) :** Cet algorithme utilise la même clé pour le chiffrement et le déchiffrement. C'est aussi la technique la plus rapide utilisée pour la transmission de données en masse. Par exemple, les deux hôtes qui doivent échanger des données confidentielles disposent tous les deux d'une clé identique. L'émetteur chiffre les données avec cette clé, puis les envoie au récepteur. Ce dernier déchiffre avec la même clé pour récupérer des données lisibles comme le représente la figure I.3. Exemples : le DES (Data Encryption Standard), le 3DES (3 Data Encryption Standard), le AES (Advanced Encryption System) et le RC4 (Rivest Cipher 4) comme algorithmes utilisés pour le chiffrement symétrique [10] :

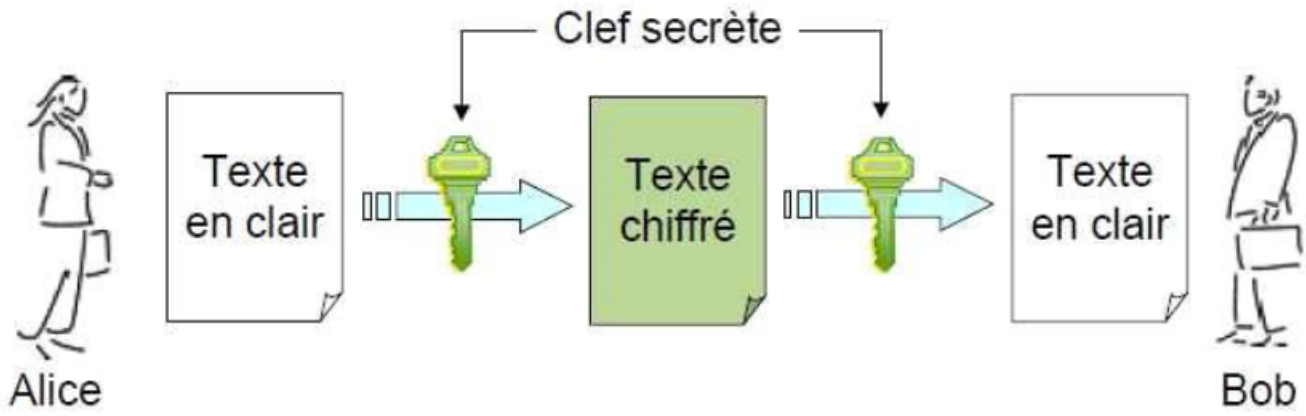

Figure I.3 – Chiffrement symétrique.

— **Chiffrement asymétrique (à clé publique) :** Ce système de cryptage utilise deux clés différentes pour chaque utilisateur : l'une est privée et n'est connue que par l'utilisateur ; l'autre est publique et donc accessible à tout le monde. Ces deux clés sont mathématiquement liées. Dans la pratique, la clé publique sert à crypter les messages, et la clé privée sert à les décrypter. Une fois le message crypté seul le destinataire est en mesure de le décrypter (voir figure I.4). Les algorithmes asymétriques les plus répandus sont : le RSA (Rivest, Shamir et Adleman) et le protocole d'échange de clés Diffie-Hellman [10].

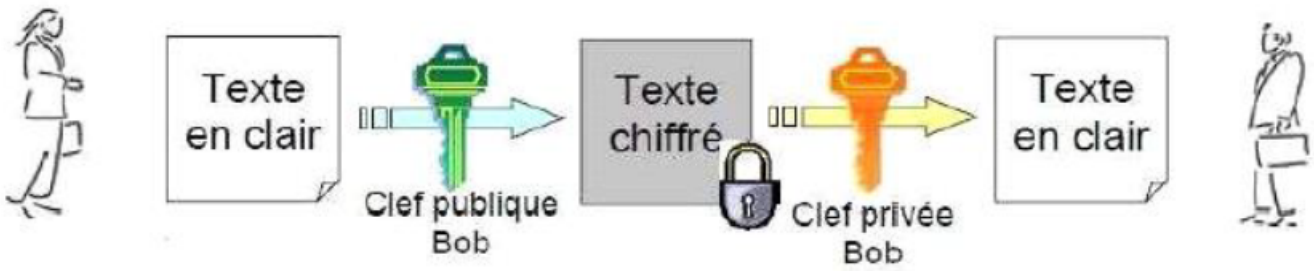

Figure I.4 – Chiffrement asymétrique.

I.2.4.2 La signature

Il est impératif, dans toute transaction professionnelle, que les deux parties fournissent une garantie de leur identité. En ce sens, la signature numérique et le certificat le permettent, car ils constituent des moyens d'identification de l'émetteur du message.

- **La signature numérique :** Le paradigme de signature numérique correspond à une marque personnelle mise sur un document électronique, qui permet l'authentification, au terme dit prouver qu'un message provient bien d'un émetteur donné. Ce mécanisme permet aussi de vérifier l'intégrité du message reçu, qui est une preuve que le document n'a pas subi

d'altération entre l'instant où il a été signé par son auteur et celui où il a été consulté, ainsi il assure la non-répudiation, afin d'éviter que l'expéditeur nie le fait d'avoir chiffré le message avec sa clé privée [11].

- **Les certificats :** Pour assurer l'intégrité des clés publiques, celles-ci sont délivrées avec un certificat. Ce dernier est une structure de données signée numériquement par une autorité certification (CA : Certification Authority). Il est composé de :
  — Nom du certificat ;
  — Informations identifiant le propriétaire de la clé publique et la clé publique elle-même ;
  — Date d'expiration et le nom de l'organisme de certificat.
  La CA utilise sa clé privée pour signer le certificat et assurer ainsi une sécurité supplémentaire. Si le destinataire connait la clé publique de la CA, il peut vérifier que le certificat provient bien de l'autorité concernée et s'assurer que le certificat contient des informations fiables et une clé publique valide [12].

I.2.4.3 Les Antivirus

Un antivirus est un logiciel conçu pour identifier et supprimer des logiciels malveillants (malware), également appelés virus, Cheval de Trois ou Ver, selon la forme. L'antivirus analyse les fichiers entrants (fichiers téléchargés ou e-mail), la mémoire vive de l'ordinateur et les périphériques de stockage tels que les disques durs, internes ou externes, les clés USB et les cartes à mémoire Flash [13]. La détection d'un logiciel malveillant peut reposer sur trois méthodes :

— Reconnaissance d'un code déjà connu (appelé signature) et mémorisé dans une base de données ;
— Analyse du comportement d'un logiciel ;
— Reconnaissance d'un code typique d'un virus.

I.3 l'authentification

I.3.1 Définition

L'authentification consiste à vérifier et a validé l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il prétend être. L'authentification complète le processus d'identification dans le sens où elle permet de prouver une identité déclarée.

I.3.2 Facteurs d'authentification

L'authentification permet l'accès aux objets du système individuellement. Elle peut être réalisée de différentes manières en se basant sur un ou plusieurs de ces facteurs :

— Les facteurs de connaissances : quelque chose que l'utilisateur connait (c'est spécifique ou secret) comme un mot de passe, une réponse à une demande d'information ou à une question secrète que peut-être d'autres ne connaissent pas ;

— Les facteurs de propriété : quelque chose que l'utilisateur possède. C'est un objet qui appartient à l'utilisateur, comme une carte à puce ou un appareil similaire ;
— Les facteurs d'inhérence : c'est un attribut physique comme les empreintes digitales ou la voix, qui peuvent être identifiées [14].

I.3.3 Méthodes et types d'authentification

Les diverses méthodes d'authentification sont représentées dans la figure suivantes [15] :

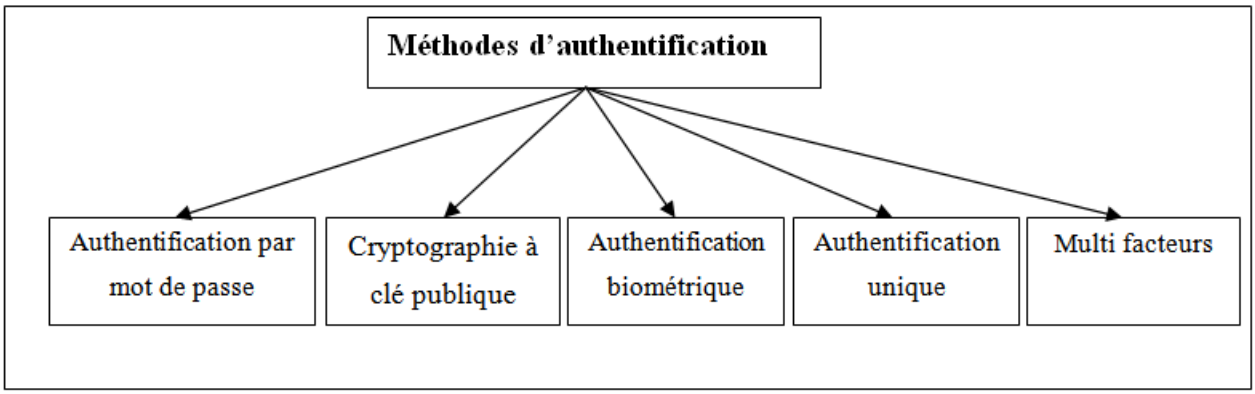

Figure I.5 – Méthodes d'authentification.

- **L'authentification simple (Authentification par mot de passe) :** elle repose sur un seul facteur d'authentification pour permettre l'accès à une ressource ;
- **L'authentification unique :** une méthode permettant à un utilisateur de s'authentifier une seule fois pour accéder à plusieurs services se trouvant sur un même système.
- **L'authentification forte (Multi facteurs) :** une procédure permettant d'identifier un utilisateur. Elle nécessite la concaténation d'au moins deux facteurs d'authentification.
- **L'authentification Biométrique :** L'authentification par biométrie s'appuie sur la vérification d'un élément du corps de l'utilisateur (le plus souvent l'empreinte digitale). Elle peut s'appuyer sur un serveur central, sur le poste ou sur une carte à puce pour stocker les données biométriques de l'utilisateur. Cette solution est en général mise en œuvre pour le processus d'authentification initiale et/ou pour protéger l'accès à des applications très sensibles [16].

I.4 Protocoles d'authentification

Plusieurs protocoles d'authentifications sont disponibles. Chaque protocole utilise certaines méthodes pour réaliser l'authentification, et ce, bien que la mise en œuvre puisse différer en termes de robustesse et de processus impliqués.

La majorité des protocoles d'authentification ont la particularité d'utiliser des secrets, soit pré-partagés ou délivrés, pour mener le processus d'authen-tification d'identité. Ils exploitent habituellement des nombres aléatoires, des fonctions de hachage, des défis et des estampilles temporelles pour améliorer la robustesse ou ajouter des fonctionnalités au protocole. Comme exemple de protocole d'authentification, nous avons : le protocole Password Authentification Protocole (PAP), Challenge Handshake Authentification Protocole (CHAP), SSL/TLS, Kerberos, RADIUS, ... etc [14].

I.5 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base de la sécurité informatique, types d'attaques et de menaces ainsi que les mécanismes qui pouvant être mis en place pour assurer la sécurité. En second lieu, nous avons énumérer les différentes techniques de l'authentification, Leurs facteurs, types, méthodes, et les protocoles d'authentifications.

Chapitre II

Le Single Sign-On et Le mécanisme CAS

II.1 Introduction

L'authentification unique (Single Sign-On ou SSO) est un processus d'authentification qui permet aux utilisateurs ou aux clients de se connecter en une seule paire login/password non seulement à un domaine donné, mais aussi de bénéficier d'une authentification automatique à un autre domaine sans autre interaction de l'utilisateur. Dans ce second chapitre, nous allons présenter, d'abord, les différents concepts de l'authentification unique (SSO).Ensuite, nous poursuivrons en présentant le protocole d'authentification (Central Authentification Service) ainsi quel 'annuaire LDAP (Lightweight Directory Access Protocol).

II.2 Définition

Le Single Sign-On (SSO) est un service d'authentification de session et d'utilisateur. Il permet à l'utilisateur d'employer un ensemble d'informations d'identification (login/password) pour accéder à plusieurs applications (Figure II.1). Le SSO peut être utilisé par les entreprises, les petites organisations et les particuliers. Il permet d'atténuer la gestion de divers noms d'utilisateurs [17].

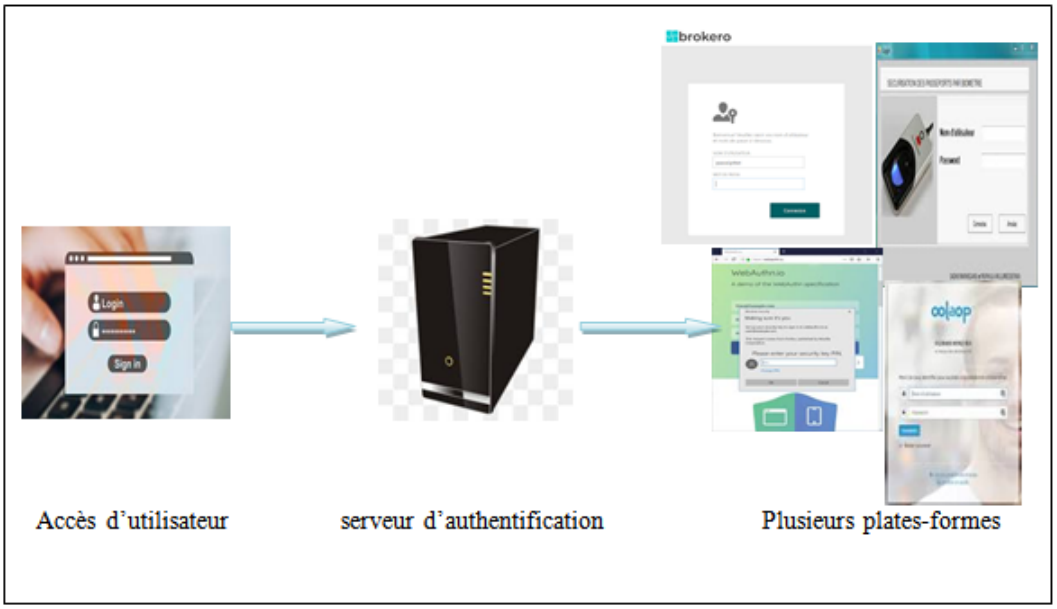

Figure II.1 – Principe du SSO.

II.3 Objectif du Single Sign-On

Le Single Sign-On permet [18] :

1. **La gestion d'un seul compte utilisateur :**
   - Le Single Sign-On unifie les comptes de l´utilisateur en un couple unique login/mot de passe. L´utilisateur s´authentifie une seule fois lors de son premier accès à l´application ;
   - L´utilisation d´un seul compte permet de résoudre le problème lié à la multiplication des mots de passe (perte ou mauvaise gestion des mots de passe par l´utilisateur). Le but est donc de centraliser les login et mots de passe des utilisateurs et d´autoriser une certaine mobilité à ces derniers ;
   - La solution offre un confort non négligeable à l´utilisateur.
2. **L'amélioration des méthodes d'authentification :**

   L´utilisation d´un service commun d´authentification permet d'optimiser les accès utilisateur aux applications du SI (système d'information). Le SSO permet notamment de mettre en œuvre une véritable politique d´authentification au niveau du réseau et des applications. Cela permet de :
   - Faciliter l´évolution des méthodes d´authentification comme le mode de transfert du login/mot de passe ou la prise en compte de plusieurs niveaux d´authentification en fonction de la sensibilité des applications accédées ;
   - Homogénéiser les connexions avec notamment les techniques de synchronisation entre domaines. Les applications pourraient être intégrées complètement dans un système

d´information, quel que soit leur type (application web, émulateur SSH). En plus d´un annuaire centralisé traditionnel, l´authentification des utilisateurs exige souvent l´utilisation de bases de données additionnelles ou de certificats. Le SSO permet de disposer d´un service d´abstraction par rapport au(x) mécanisme(s) d´authentification ;
— Gérer les droits et les autorisations d´accès à une application, car il est nécessaire pour certaines applications de pouvoir disposer d´informations définissant les rôles des utilisateurs.

II.4 Avantages et inconvénients du Single Sign-On

Avantages [19] :

— Une limitation des risques en matière de sécurité :les utilisateurs finaux n'ont plus besoin de noter leurs noms d'utilisateurs et leurs mots de passe sur une feuille qu'ils laisseraient à proximité de leur ordinateur ;
— Une meilleure sécurité du réseau d'entreprise : la fonctionnalité SSO empêche les utilisateurs d'accéder sans autorisation au réseau de l'entreprise. Ils peuvent accéder uniquement aux applications pour lesquelles ils ont une autorisation ;
— Une amélioration du niveau de service : comme le service support informatique reçoit moins d'appels relatifs aux mots de passe, ses collaborateurs peuvent se concentrer sur les appels à caractère plus critique et améliorer ainsi leur niveau de service.

Inconvénients :

Malgré les avantages majeurs qu'apporte le SSO, il est important de soulever les quelques inconvénients ci-après [20] :

— Intégration complète difficile dans le SI (Système d'Information). Cette difficulté réside dans la mise en conformité du SI de l'entreprise ;
— Coût et lourdeur : l'affiliation d'une application à un système SSO a un coût non négligeable en termes de budget et engendre une lourdeur au niveau des accès serveur ;
— Le SSO peut également nuire à la sécurité, car il donne accès à une multitude de ressources une fois l'utilisateur authentifié. Raison pour laquelle, il est préférable de coupler les solutions de SSO avec un système d'authentification fort. Aussi, si une panne survient sur le serveur d´authentification, celle-ci touchera également l´application de SSO. Il est donc préférable de mettre en place un serveur de secours.

II.5 Architecture d'un SSO

L'architecture de la plupart des produits de SSO est inspirée de Kerberos. On y utilise largement sa terminologie et ces produits partagent ses concepts de base qui sont les suivants [21] :

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).

- **Le serveur d'authentification :**
  Le serveur d'authentification est l'élément central du système de SSO. Il assure l'authentification de l'utilisateur qui lui fournit ses éléments d'authentification. Si le mode d'authentification est le mot de passe, la phase d'authentification implique la vérification de celui-ci auprès d'une base de référence. La plupart des systèmes de SSO implémentent plusieurs backend[1] d'authentification (/etc./Password, NIS, LDAP) [22].
- **L'agent d'authentification :**
  L'agent vérifie que l'utilisateur est authentifié ; s'il ne l'est pas, il le redirige vers le serveur d'authentification. Si le client est déjà authentifié auprès du serveur d'authentification, le serveur le redirige directement vers l'agent d'authentification demandeur [21].
  Les principaux modèles d'architecture SSO sont [23] :
  1. SSO côté client (client-Side SSO).
  2. SSO côté serveur (server-Side SSO).
  3. Des systèmes hybrides.

[1]

II.5.1 SSO côté client :

Le SSO côté client est présenté comme suite (figure II.2) :

1. [1]**Backend :** est la partie du code qui est exécutée par le serveur, il s'agit du travail qu'il réalise sur les pages Web des sites dynamique avant de les envoyer au client.

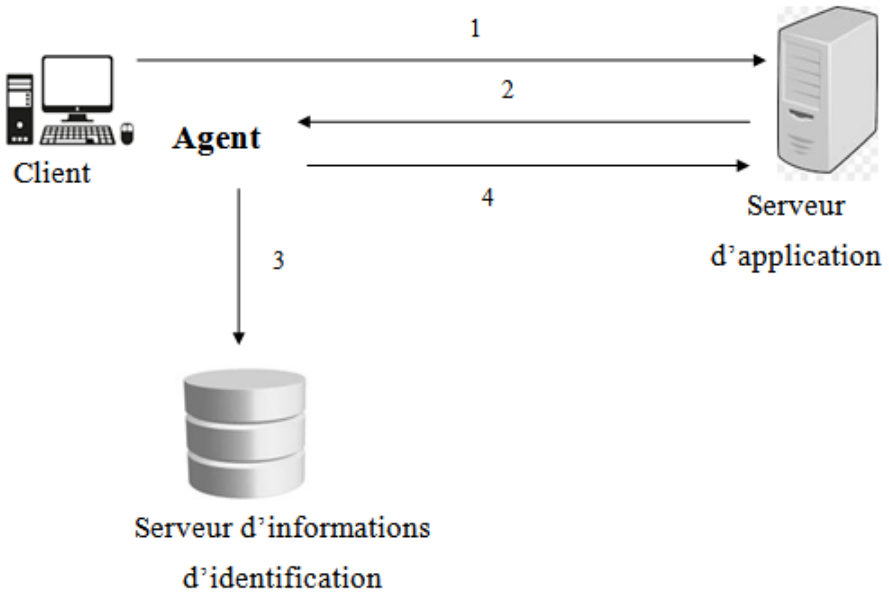

Figure II.2 – SSO coté client avec serveur d'informations d'identification.

1. Le client souhaite accéder à une application.
2. L'application demande ses informations d'identification. L'agent présent sur le poste client intercepte la demande.
3. L'agent vérifie les informations d'identification dans le serveur centralisé des identifications.
4. L'agent simule l'utilisateur réel et envoie les informations d'identification à l'application. Ainsi, l'identification est transparente pour l'utilisateur.

II.5.2 SSO côté serveur :

Deux types de SSO « Server Side » existent : ceux qui utilisent un **"Reverse Proxy"** et ceux utilisant des agents **"Serveurs"** [23].

II.5.2.1 Architecture avec agent serveur :

La figure suivante présente l'architecture avec agent serveur (figure II.3) :

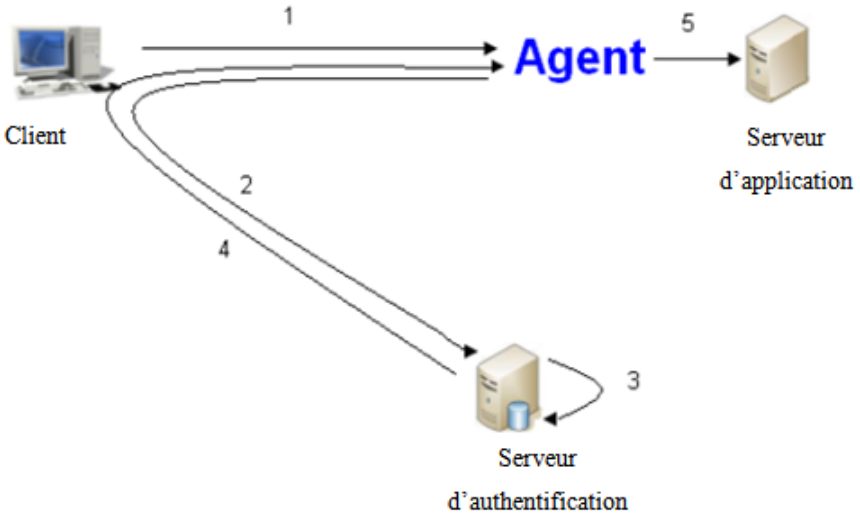

FIGURE II.3 – SSO coté client avec un agent de serveur d'authentification.

1. Le client souhaite accéder à une application. L'agent présent sur le serveur d'application intercepte la demande.
2. L'agent vérifie que l'utilisateur est authentifié : s'il ne l'est pas, l'agent se trouvant au niveau du client redirige la requête vers le serveur d'authentification. Cette redirection apparait sous forme d'un portail ou d'une fenêtre. L'utilisateur fournit ses informations d'identification pour le serveur d'authentification.
3. Le serveur d'authentification vérifie l'identité de l'utilisateur dans la base de données de référence.
4. Une fois l'utilisateur authentifié, le serveur d'authentification renvoie un cookie[2] HTTP sur le poste de l'utilisateur qui permet de maintenir la session de l'utilisateur.
5. L'agent le transfert sur le serveur d'application.

[2]

II.5.2.2 Architecture avec Reverse Proxy :

Cette architecture est présentée comme suite (figure II.4) :

2. [2]**cookie :** est un fichier qui permet d'enregistrer des informations relative à la navigation d'un ordinateur sur le site internet.

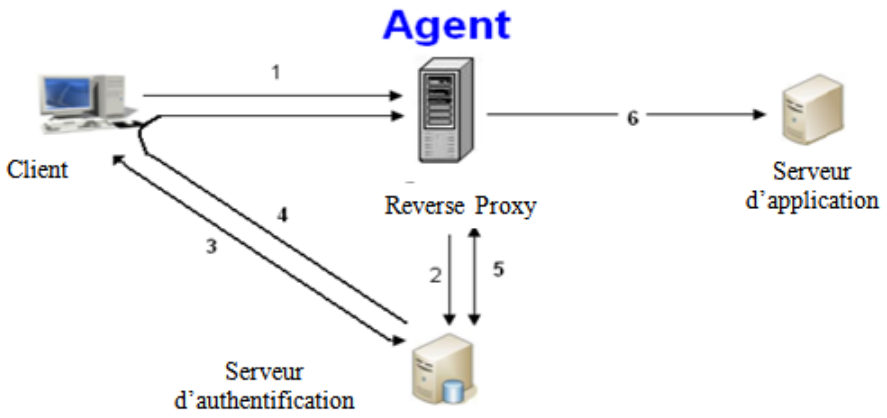

Figure II.4 – SSO coté client serveur avec agent de reverse proxy.

1. L'utilisateur tente de se connecter à l'application Web. Toute demande de connexion pour une application est redirigée vers le reverse proxy.
2. L'agent sur le reverse proxy intercepte la demande et vérifie l'authentification de l'utilisateur via le serveur d'authentification.
3. Si l'utilisateur n'est pas authentifié, le serveur d'authentification demande à l'utilisateur des informations d'identification. L'utilisateur fournit ses informations d'identification pour le serveur d'authentification.
4. Le serveur d'authentification envoie un jeton jouant le rôle de cookie et redirige le navigateur.
5. L'agent sur le reverse proxy intercepte la demande et vérifie l'authentification de l'utilisateur via le serveur d'authentification avec le jeton. Le serveur d'authentification envoie le login et l'autorisation des informations qui est associée avec le jeton.
6. L'agent permet d'accéder à la demande.

II.5.3 SSO hybrides :

Les approches dites hybrides combinent le client-side SSO et le server-side SSO et elles sont nombreuses. L'avantage de cette architecture est qu'elle permet de réduire les problèmes du SSO côté client [23].

II.6 Les différentes approches du SSO

II.6.1 L'approche centralisée :

Le principe de base ici est de disposer d'une base de données globale et centralisée de tous les utilisateurs ou d'un annuaire (figure II.5). Cela permet de centraliser la gestion de la politique de sécurité. Un exemple de mise en œuvre est le logiciel libre Lemon LDAP, un autre exemple est le logiciel libre Vulture [22].

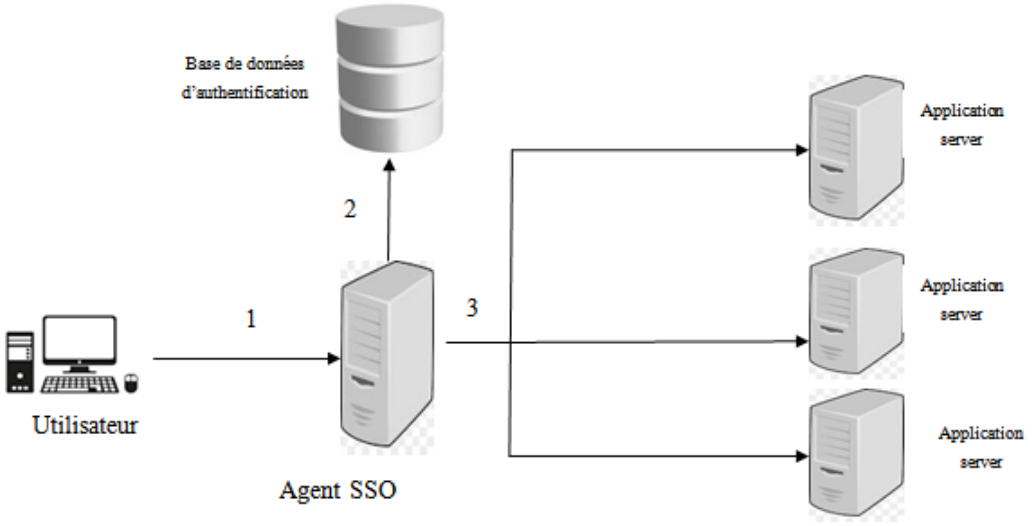

Figure II.5 – SSO centralisé.

1. L'utilisateur souhaite accéder à une application. Dans ce cas, l'agent qui s'exécute sur un Reverse Proxy intercepte la demande.
2. L'agent authentifie l'utilisateur dans une base de données d'authentification qui peut être un annuaire LDAP.
3. Une fois l'utilisateur authentifié, il peut accéder à l'application.

II.6.2 L'approche fédérative :

Dans cette approche, chaque service gère une partie des données d'un utilisateur (l'utilisateur peut donc disposer de plusieurs comptes), mais partage les informations dont il dispose sur l'utilisateur avec les services partenaires, exemple : le système Liberty Alliance[3] .

Cette approche a été développée pour répondre à un besoin de gestion décentralisé des utilisateurs où chaque service partenaire désire conserver la maîtrise de sa propre politique de sécurité (figure II.6). En exemple, un ensemble de sites marchands indépendants d'un point de vue commercial et organisationnel [22].

3

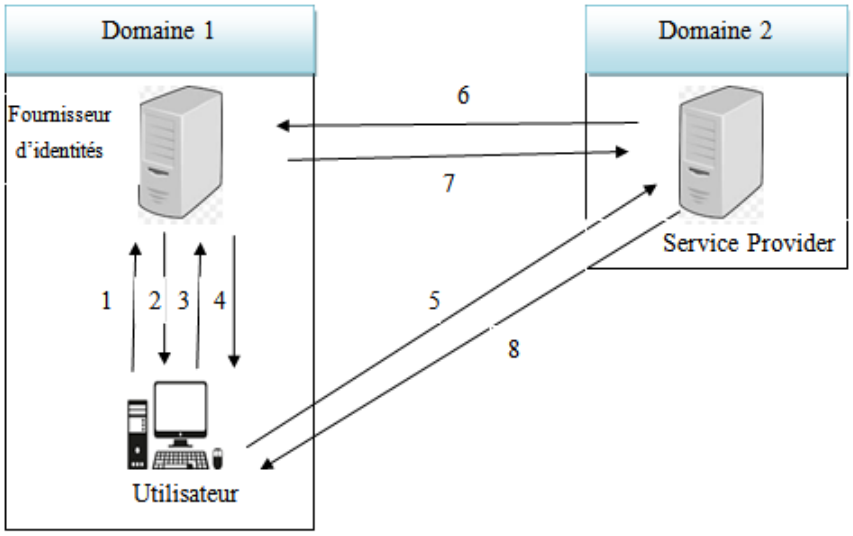

Figure II.6 – La fédération du SSO.

1. L'utilisateur se connecte au fournisseur d'identité.
2. Une fois l'authentification réussie, l'IDP (Identifier Provider) envoie à l'utilisateur des informations sur les applications auxquelles il peut accéder.
3. L'utilisateur clique sur le lien Service Provider (SP) dans le portail. Il s'agit d'un lien spécial qui ne se connecte pas directement au SP.
4. L'IDP reçoit la demande et crée une « Identité Assertion » (un identifiant d'identité). L'IDP conserve cette « Identité Assertion » avec un "artefact" pointeur dans son cache. Puis, l'IDP renvoie une réponse redirigée vers le navigateur client.
5. Le navigateur est redirigé vers le SP avec «l'artefact».
6. Le SP reçoit cette demande et contacte l'IDP avec « l'artefact » pour demander si l'identité est confirmée.
7. L' IDP reçoit la demande, et vérifie cette entrée dans la table des « Identity Assertion » en cache en utilisant « l'artefact » comme index.

3. [3]**Liberty Alliance :** ou le Project Liberty, est un projet qui réunit des acteurs des mondes industriel, informatique, ... etc, pour objectif de définir des ensembles de spécifications de protocoles de fédération d'identité et de communication entre site web

II.6.3 L'approche coopérative :

L'approche coopérative, dont les systèmes Shibboleth et Central Authentication Service sont les principaux représentants, part du principe que chaque utilisateur dépend d'une des entités partenaires. Ainsi, lorsqu'il cherche à accéder à un service du réseau, l'utilisateur est authentifié par le partenaire dont il dépend, comme dans l'approche fédérative. Cependant, chaque service du réseau gère indépendamment sa propre politique de sécurité. Cette approche est principalement utilisée par les communautés universitaires [22].

II.7 Les solutions d'un SSO existantes

Le tableau II.1 illustre les différentes implémentations d'un SSO existantes dans le commerce :

Le Produit	Descriptions
Solution SAML	SAML pour Security Assertion Markup Language permet entre autres la délégation d'authentification et sert de fondation à deux autres normes, Shibboleth et Liberty Alliance.
Solution SHIBBOLETH	Shibboleth est un mécanisme de propagation d'identités développé par le consortium Internet2, qui regroupe 207 universités et centres de recherches. La propagation d'identités est la délégation de l'authentification à l'établissement d'origine de l'utilisateur et l'obtention de certains attributs de ce dernier.
Solution OpenID	Implémentée et utilisée par les sociétés clées de l'Internet (Yahoo, MySpace, Google, Microsoft...), elle propose un protocole ouvert pour une gestion décentralisée d'identités mettant l'utilisateur au centre des décisions le concernant.
Solution Liberty Alliance	L'authentification unique dans le contexte Liberty Alliance correspond à la possibilité pour l'utilisateur d'accéder, après s'être identifié à l'aide d'un compte unique, à des services proposés par différents fournisseurs appartenant à un même « cercle de confiance ».
Solution CAS	CAS pour Central Authentication Service : s'authentifie sur une application et on est alors authentifié sur toutes les applications qui utilisent le même serveur CAS.

Table II.1 – Solution du SSO [2]

.

II.8 CAS (Central Authentification Service) :

II.8.1 Définition

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 de données, etc... L'architecture est générique et ne dépend pas du système d'authentification choisi [24].

II.8.2 Mécanisme de CAS

II.8.2.1 L'architecture de CAS

L'architecture CAS repose sur trois acteurs principaux :

- **CAS server :** le serveur CAS est l'élément central de l'authentification. Ce serveur est le seul acteur du mécanisme CAS qui relie les mots de passe des utilisateurs vers un annuaire LDAP centralisé. Son rôle est de [21] :
  - Authentifier les utilisateurs ;
  - Certifier l'identité de la personne authentifiée aux CAS par son ticket de service.
- **CAS client :** un client CAS est tout fournisseur de services compatible avec CAS et pouvant communiquer avec le serveur. C'est un progiciel qui peut être intégré à plusieurs plates-formes logicielles et applications afin de communiquer avec serveur CAS [24].
- **Les navigateurs Web :** Ils doivent satisfaire les contraintes suivantes pour bénéficier de tout le confort de CAS [21] :
  - Permettre le protocole HTTPS ;
  - 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).

II.8.2.2 Fonctionnement de base de CAS

1. **Authentification d'un utilisateur :** un utilisateur non authentifié, ou dont l'authentification a expiré, qui accède au serveur CAS se voit proposer un formulaire d'authentification (Figure II.7), dans lequel il est invité à entrer ses informations d'authentifications (login/email et mot de passe [1].

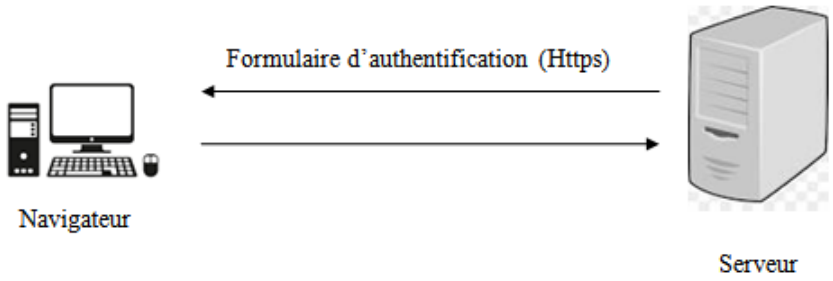

Figure II.7 – Premier accès d'un navigateur au serveur CAS.

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

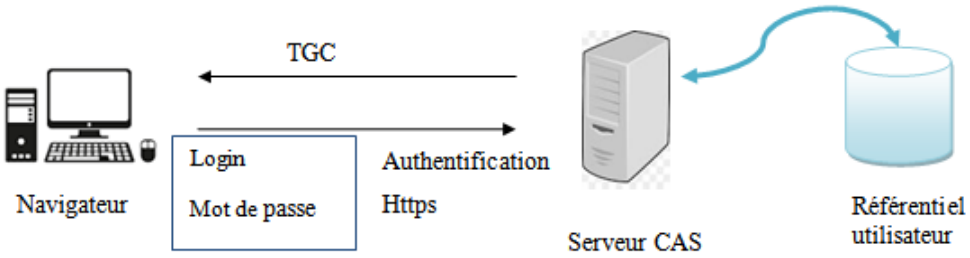

Figure II.8 – Authentification d'un navigateur auprès du serveur CAS.

Ticket Granting Cookie (TGC) : est le passeport de l'utilisateur auprès du serveur CAS. Ayant une durée de vie limitée, TGC est le moyen pour les navigateurs d'obtenir auprès du serveur CAS des tickets de services sans avoir à se ré authentifier. C'est un cookie privé et protégé. Il est aussi opaque tout comme les autres tickets utilisés dans le mécanisme CAS [25].

2. **Accès à une ressource protégée après authentification :** la figure ci-dessous représente l'ordre d'accès aux ressources protégées par CAS :

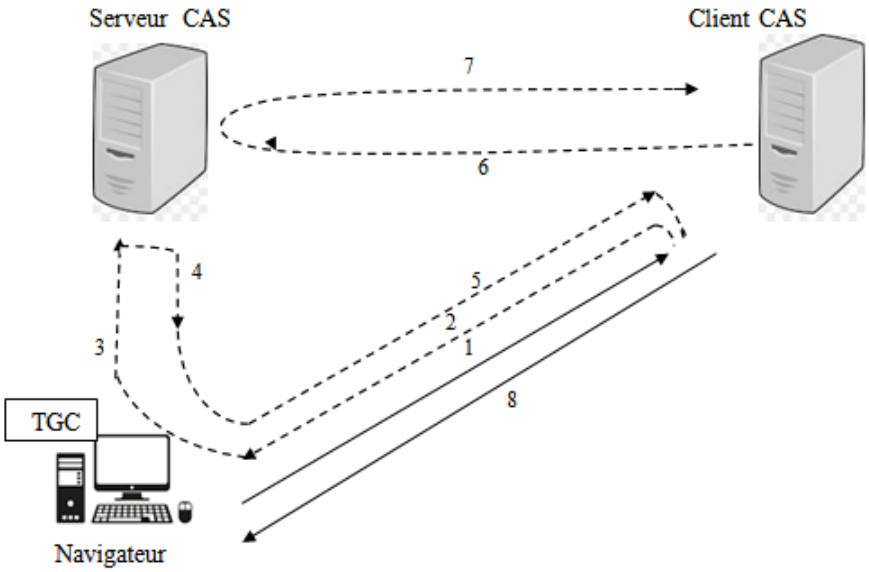

Figure II.9 – Redirection par le serveur CAS d'un navigateur vers un client CAS après authentification.

Les étapes sont [26] :

1- Le navigateur tente d'accéder à une ressource protégée sur un client CAS ;

2- Le client redirige le navigateur vers le serveur CAS dans le but d'authentifier l'utilisateur ;

3- Le navigateur, précédemment authentifié auprès du serveur CAS, lui présente le TGC ;

4- Dès que le TGC apparait sur le serveur CAS, celui-ci émet un Service Ticket (Service Ticket ou ST) au navigateur ; il s'agit d'un ticket opaque qui ne comporte aucune information personnelle ; il n'est accessible que par le « service » (L'URL) le demandant ;

5- Dans le même temps, le serveur CAS redirige le navigateur vers le service demandeur en passant le ST en paramètre ;

6- Et 7- Le ticket de service est ensuite validé par le client CAS directement en http avec le serveur CAS ;

8- La ressource est livrée au navigateur.

3. **Accès à une ressource protégée sans authentification préalable :**
   Si l'utilisateur n'est pas déjà authentifié auprès du serveur CAS avant d'accéder à une ressource, son navigateur est, comme précédemment, redirigé vers le serveur CAS qui lui propose alors un formulaire d'authentification [22]. Lors de la soumission du formulaire par le navigateur au serveur CAS, si les informations fournies sont correctes, le serveur CAS [2] :
   — Emet un TGC au client, qui lui permettra ultérieurement de ne pas avoir à se re-authentifier ;
   — Délivre au client un Service Ticket à destination du client CAS ;
   — Redirige le client vers le client CAS.

II.8.2.3 Fonctionnement multi-tiers :

1. **Les mandataires (Proxies) CAS :** Pour les clients CAS, le multi-tiers de CAS fonctionne comme un navigateur. Un tel client CAS est appelé proxy CAS. Exemple [1] :
   — Le portail Web sur lequel l'utilisateur s'est authentifié peut avoir besoin d'interroger des applications externes en tant qu'utilisateur se connectant (service web) ;
   — La passerelle de messagerie Web authentifiée de l'utilisateur (web mail) doit se connecter à un serveur IMAP pour récupérer l'e-mail de l'utilisateur sous son identité.
2. **Le fonctionnement multi-tiers :** Le proxy CAS émet une requête PGT (Proxy Granting Ticket) en même temps qu'il valide un Ticket Service (TS) pour authentifier un utilisateur (Figure II.10) :

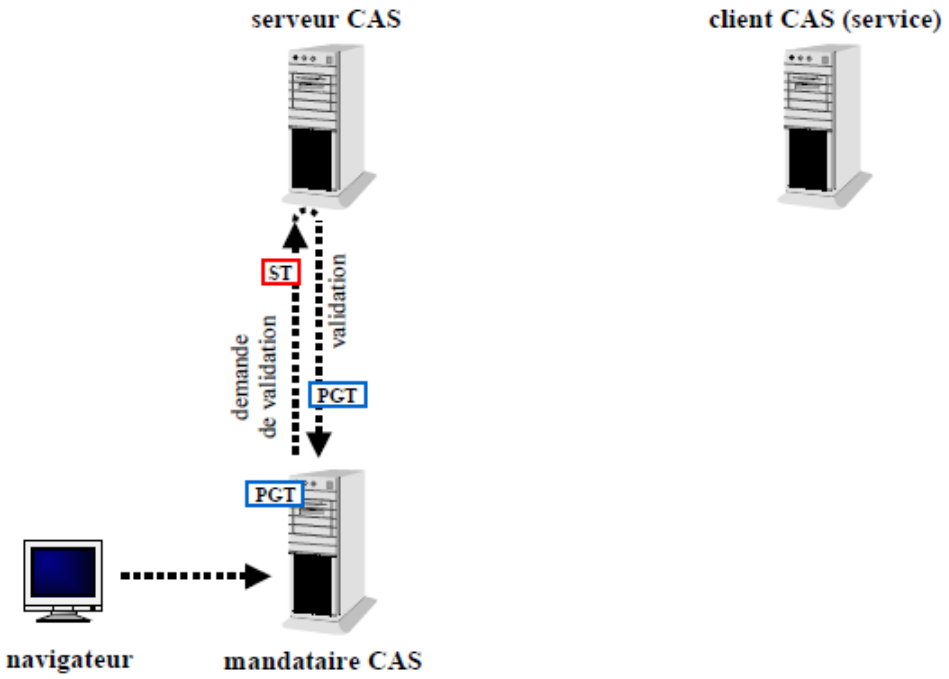

Figure II.10 – Récupération d'un PGT par un mandataire CAS auprès du serveur CAS [1]

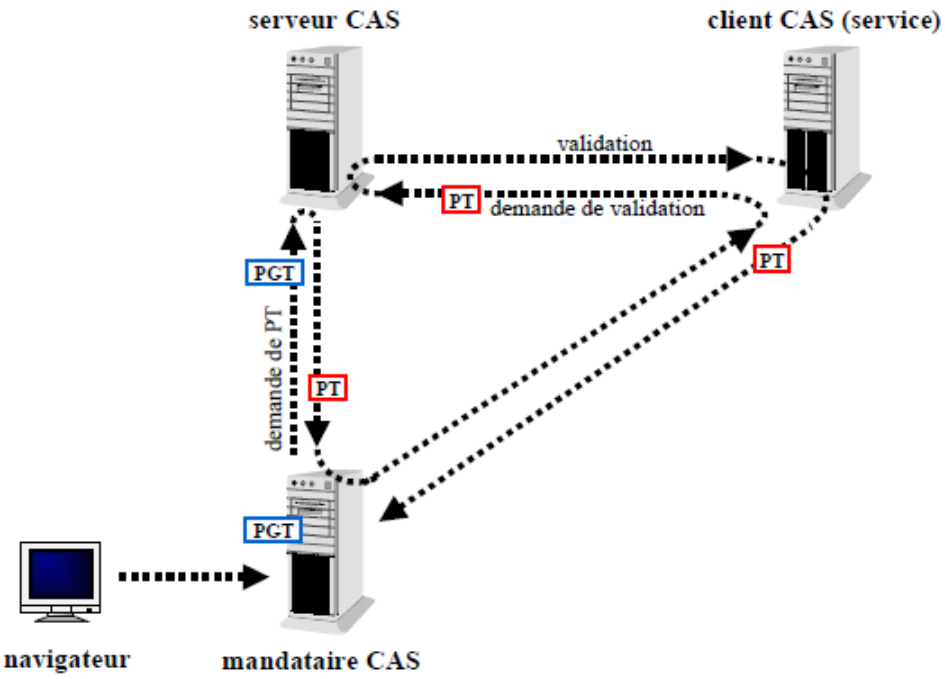

Figure II.11 – Validation d'un PT par un client CAS accédé par un mandataire CAS [31].

- **Le Proxy Granting Ticket :** est le passeport d'un mandataire CAS, pour un utilisateur, auprès du serveur CAS. Le PGT est opaque, et obtenu du serveur CAS par un canal chiffré. Comme le TGC, sa durée de vie est très limitée [1].
- **Les Proxy Tickets (PT) :** sont, comme les Service Tickets, validés par le serveur CAS pour donner accès à des ressources protégées [1].

Le client CAS accédé par le mandataire CAS du fonctionnement 2-tiers peut également être mandataire à son tour. Les mandataires peuvent ainsi être chaînés (Figure II.12). CAS est le seul mécanisme de SSO proposant un tel fonctionnement multi-tiers sans aucune propagation de mot de passe [1].

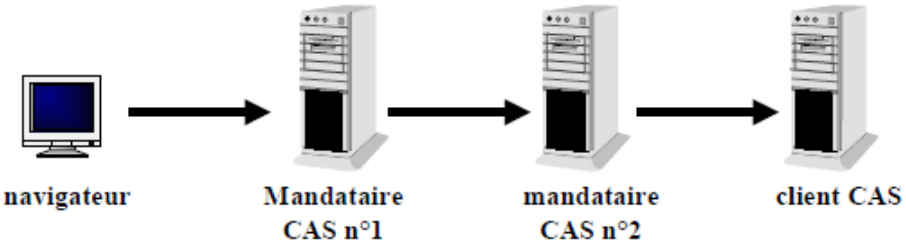

Figure II.12 – Fonctionnement de n-tiers.

II.8.2.4 Avantages de CAS [27] :

— Une meilleure sécurité des mots de passe : CAS atténue le risque de compromettre les mots de passe du compte perdu lié au serveur CAS ;
— Une expérience d'authentification utilisateur plus cohérente. Chaque application Web exploitant CAS utilise le même écran de connexion à partir de la même URL, Ce qui rassure les utilisateurs ;
— Une extensibilité facile : il permet aux serveurs Web de bénéficier immédiatement de méthodes d'authentification supplémentaires ;
— Un meilleur support utilisateur : CAS permet une assistance centralisée à l'authentification du compte perdu, disponible via le centre de service client.

II.9 Authentification avec l'annuaire LDAP

LDAP ou Lightweight Directory Access Protocol est un protocole ouvert et multiplateforme utilisé pour l'authentification des services d'annuaire. LDAP fournit le langage de communication utilisé par les applications pour communiquer avec d'autres serveurs de services d'annuaire [28].

II.9.1 Les annuaires LDAP :

Les principaux annuaires LDAP existant sur le marché [24] :
— Open LDAP ;
— Apache Directory Server ;
— Sun (One/Java) Directory Server ;
— Active Directory.

II.9.2 Utilité de LDAP :

Par sa capacité à s'adapter à plusieurs cas d'utilisation, en plus de l'authentification, un annuaire LDAP devient pratiquement indispensable au cœur d'un système informatique. Il est utilisé notamment pour [29] :
— Les outils de messagerie ont comme fonctionnalité d'interroger un annuaire LDAP pour tirer leur carnet d'adresse ;
— Les solutions d'infrastructure de sécurité (Firewall, proxy, etc.) peuvent y lire leurs paramétrages ;
— Une majorité des solutions logicielles du marché (gestion de contenu, portails d'entreprise, progiciels de gestion, etc) utilisent LDAP pour l'authentification.

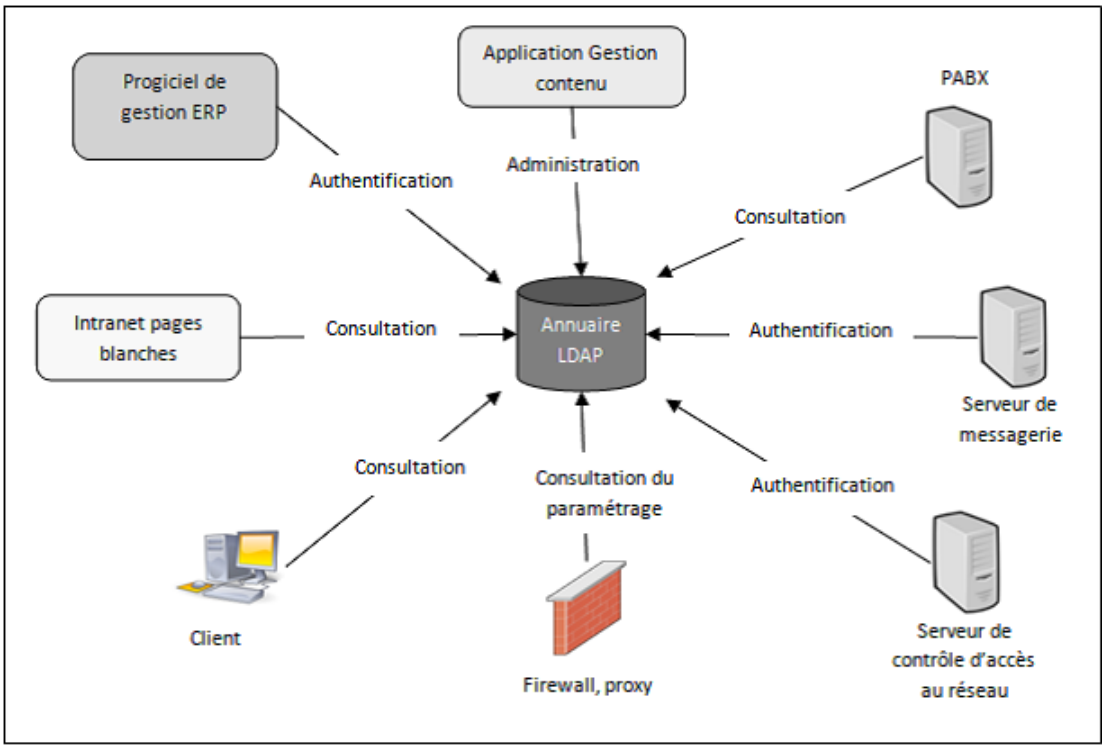

Figure II.13 – LDAP au cœur d'un système.

L'annuaire LDAP fournit les services suivants :

— Un protocole réseau pour accéder à l'information contenue dans un annuaire ;
— Un modèle d'information définissant la forme et le type de l'information ;
— Un espace de nommage définissant comment l'information est organisée et référencée ;
— Un modèle de distribution permettant de diffuser les données ;
— Un protocole et un modèle de données extensible [19].

II.10 Conclusion

Le Single Sign-On est un élément fédérateur pour les applications web, avec la montée en charge des applicatifs et le nombre croissant d'utilisateurs, l'attribution de comptes d'utilisateurs et la gestion des accès deviennent de plus en plus importantes.

Chacune des solutions d'authentification présentée plus haut se distingue des autres par ses particularités et ses avantages. Étant donné la particularité impressionnante des deux solutions CAS et LDAP, nous les avons sélectionné et associé ensemble pour établir notre système d'authentification SSO.

Chapitre III

Mise en œuvre

III.1 Introduction

Dans ce présent chapitre, consacré à la mise en œuvre et à la réalisation de notre projet, nous allons présenter les outils adoptés à l'aboutissement du'un aperçu général de notre l'application.

III.2 Présentation du projet

Le nombre croissant des applications Web dans les différents domaines engendre le surnombre des mots de passe. Le présent projet vient pallier cette contrainte. Il consiste ainsi à trouver une solution au problème de l'authentification des applications Web.
Notre travail se résume en l'implémentation d'un système d'authentification unique basé sur un annuaire centralisé. Cette solution va permettre aux utilisateurs d'un réseau d'accéder, en toute transparence, à l'ensemble des ressources autorisées. Afin d'aboutir à notre objectif, nous avons, d'une part, utilisé les méthodes et les outils déjà décrits dans les chapitres précédents (annuaire LDAP, CAS-SSO) et d'autre part, intégré le service Web Java-Remote Method Invocation (RMI) pour l'interface de l'authentification. La figure (III.1) représente l'architecture sur laquelle notre solution est basée :

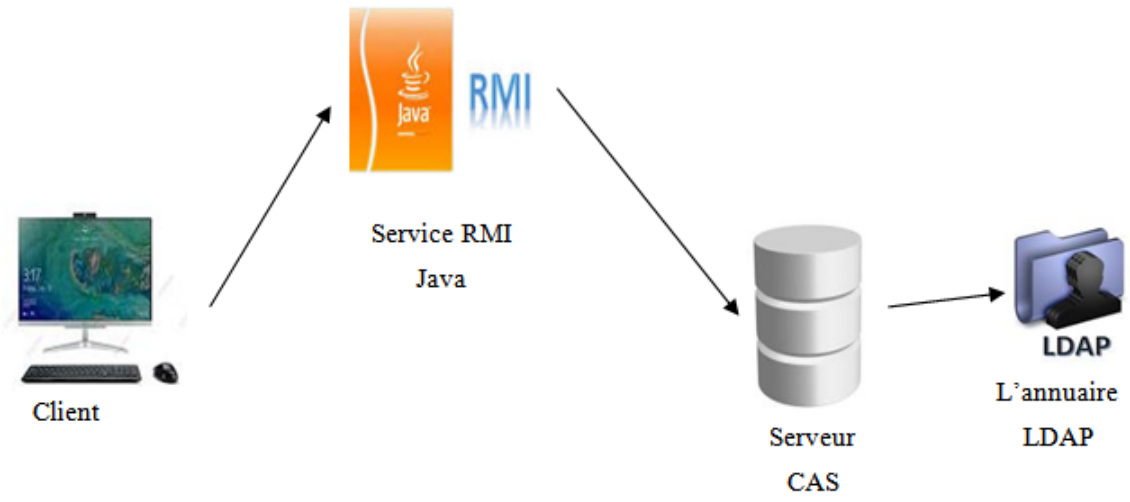

Figure III.1 – L'architecture du système.

L'architecture est composée de :

- Un annuaire LDAP qui contient la liste de tous les utilisateurs avec la description de leurs droits d'accès.
- Un serveur CAS qui va assurer l'interrogation de l'annuaire LDAP, l'authentification des utilisateurs et la propagation de session.
- Une application Java-RMI qui contient l'interface de l'authentification.

III.3 Installation et configuration des serveurs

Pour l'implémentation de notre architecture, nous avons opté pour une machine virtuelle dans laquelle nous installons Debian 11 qui est un système d'exploitation universel ; une excellente distribution de GNU/Linux open source, le système Debian est bien développé et reconnu comme excellent corps d'ingénierie logicielle. Cette machine va contenir le serveur d'authentification unique Apereo-CAS et l'annuaire LDAP .

III.3.1 OpenLDAP

OpenLDAP est une implémentation libre du protocole LDAP et un annuaire informatique qui fonctionne sur le modèle client/serveur. Il contient des informations qui sont rangées de manière hiérarchique. En pratique, dans un réseau informatique, OpenLDAP est utilisé pour enregistrer

une grande quantité d'utilisateurs ou de services. Il permet d'organiser hiérarchiquement les utilisateurs par département, par lieu géographique ou par autre critère [30].

Afin de mettre en fonction le serveur OpenLDAP, nous allons commencer par configurer le service DNS (Domain Name System) qui est un service TCP/IP permettant la correspondance entre un nom de domaine qualifié (FQDN : Fully Qualified Domain Name) et une adresse IP.

III.3.1.1 Installation et configuration

Etape 1 : DNS (Domain Name System)

Pour la configuration du DNS, nous allons utiliser la commande suivante pour installer les paquets nécessaires :

apt-get install bind9 bind9-doc bind9-utils

- On édite notre fichier de zone directe **"domain.univ "** avec la commande suivante :

nano /etc/bind/db.domain.univ

- On modifie le fichier pour avoir ceci :

```
@       IN      SOA     debian.domain.univ. root.domain.univ. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      domain.univ.
@       IN      A       192.168.36.108
debian  IN      A       192.168.36.108
```

Figure III.2 – Configuration du nom de domaine.

- On édite ensuite notre fichier **" named.conf.default-zones "** afin de déclarer notre zone avec la commande suivante :

nano /etc/bind/named.conf.default-zones

-on ajoute les lignes suivantes à la fin du fichier :

```
zone domain.univ {
        type master ;
        file /etc/bind/db.domain.univ ;
};
```

- Pour redémarrer le service **"bind9 "**, on utilise la commande :

systemctl restart bind9

- Pour finir, on modifie notre fichier **" resolv.conf "** avec la commande suivante :

nano /etc/resolv.conf

- On modifie les lignes du fichier pour avoir ceci :

search domain.univ
nameserver 192.168.36.108

- Après avoir terminé la configuration du DNS, on va tester la résolution de **" debian.domain.univ "** avec la commande suivante :

nslookup debian.domain.univ

- Le résultat ci-dessous montre que le service DNS a été bien configuré :

```
Server:         192.168.36.108
Address:        192.168.36.108#53

Name:   debian.domain.univ
Address: 192.168.36.108
```

Figure III.3 – Teste de la configuration du DNS.

Etape 2 : Le serveur OpenLDAP

- L'installation des paquets d'OpenLDAP se fait avec la commande suivante :

apt-get install slapd ldap-utils

- Pour démarrer LDAP, on utilise la commande :

systemctl start ldap

- Pour le redémarrer :

systemctl restart ldap

- Pour l'arrêter :

systemctl stop ldap

- Après avoir installé le serveur OpenLDAP, nous allons le reconfigurer avec la commande :

sudo dpkg-reconfigure slapd

- Une fois la reconfiguration de LDAP terminée, nous allons utiliser cette commande **" sudo slapcat "** qui devrait nous fournir le résultat suivant :

```
dn: dc=domain,dc=univ
objectClass: top
objectClass: dcObject
objectClass: organization
o: domain.univ
dc: domain
structuralObjectClass: organization
entryUUID: 8dce4d98-7f5c-103c-806d-6d3724c2b700
creatorsName: cn=admin,dc=domain,dc=univ
createTimestamp: 202206131203l7Z
entryCSN: 20220613120317.046427Z#000000#000#000000
modifiersName: cn=admin,dc=domain,dc=univ
modifyTimestamp: 202206131203l7Z
```

Figure III.4 – Résultat de la configuration de LDAP.

- L'étape suivante consiste à ajouter le nom de domaine de base pour les utilisateurs et les groupes. Nous allons créer un fichier nommé **" basedn.ldif "** avec le contenu ci-dessous :

dn : ou=people,dc=domain,dc=univ
objectClass : organizationalUnit
ou : people
dn : ou=groups,dc=domain,dc=univ

- Une fois terminé, nous allons appliquer les configurations à l'annuaire avec la commande suivante :

sudo ldapadd –x –D cn=admin,dc=domain,dc=univ –W –f basedn.ldif

- L'ajout des comptes des utilisateurs et des groupes se fait dans le fichier **" ldapusers.ldif "** et **" ldapgroups.ldif "** respectivement.

Etape 3 : Installation du gestionnaire de compte LDAP

À présent, nous allons installer et utiliser LDAP Account Manager comme tableau de bord de gestion graphique du serveur OpenLDAP.

LDAP Account Manager (LAM) est une interface Web pour la gestion des entrées (utilisateurs, groupes) stockées dans un annuaire LDAP. L'outil LDAP Account Manager a été créé pour rendre la gestion LDAP accessible pour l'utilisateur. LAM facilite l'administration des entrées LDAP en résumant les détails techniques de LDAP et en accordant aux administrateurs et aux utilisateurs de gérer le serveur LDAP. [31].

- L'installation des paquets se fait par les deux commandes suivantes :

apt-get install ldap-account-manager

apt-get install ldap-account-manager-lamdaemon

- Une fois l'installation des paquets, nous allons accéder à l'interface Web du gestionnaire de compte LDAP (figure III.5).

Dans notre cas nous utilisons : **http ://192.168.36.108/lam/** :

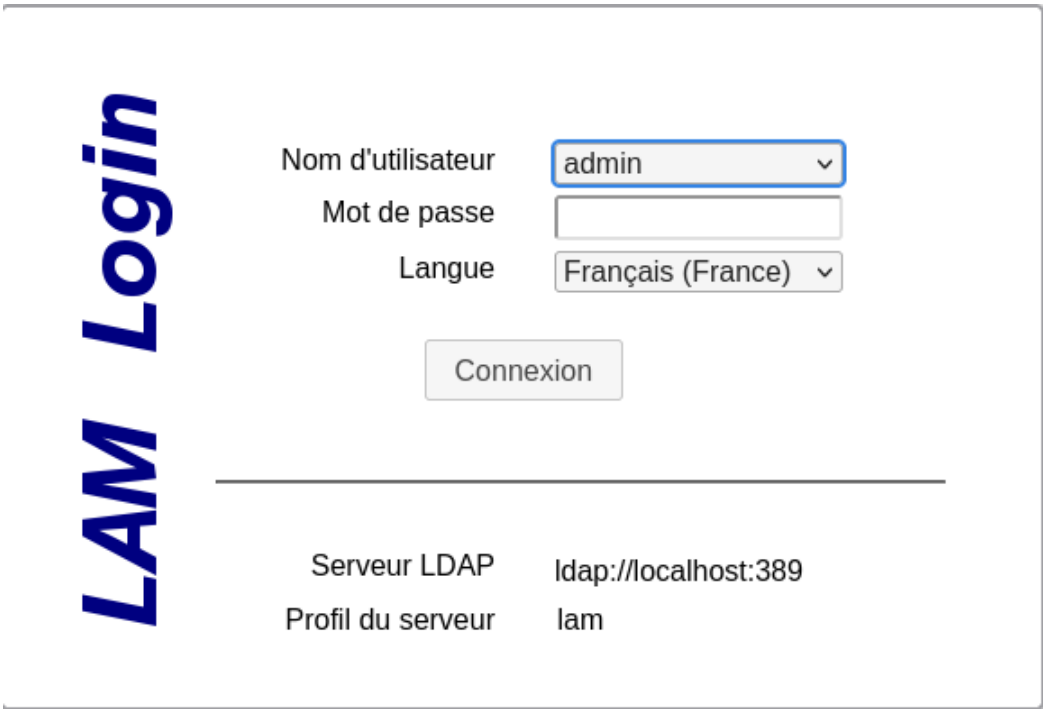

Figure III.5 – Interface Web de LDAP.

III.3.2 Installation du serveur d'authentification unique Apereo-Central Authentification Server (CAS) 6

A. Apache tomcat :

Apache-Tomcat est le serveur d'application Java du projet Jakarta de la fondation Apache. Il permet la mise en œuvre de Java Servlets et de Java Server Pages (JSP) pour promouvoir un environnement de serveur Java efficace, et exécute également un serveur Web http [32].

B. Maven :

Maven est un outil d'open source développé par la fondation Apache. Il permet de faciliter et d'automatiser certaines taches de la gestion d'un projet Java, et est nécessaire pour mettre en place CAS qui est un projet développé en Java. Maven est utilisé pour compiler le code source du projet CAS et renvoie un fichier (.war) qui est le servlet de déploiement pour le serveur CAS [33].

Etape 1 : Installation et configuration des paquets

- L'installation se fait par la commande suivante :

apt-get install tomcat9 tomcat9-admin tomcat9-user openjdk-11-jdk openjdk-11-jre maven build-essential git

- Nous configurons la variable d'environnement Java à l'aide des deux commandes :

echo JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64/ » /etc/environment

source/etc/environment

- Pour vérifier si la variable d'environnement Java a été bien ajoutée, nous utilisons la commande :

echo $JAVA_HOME

- Pour configurer Tomcat9, nous ouvrons le fichier tomcat9 qui se trouve dans le dossier **" /etc/default "** et nous ajoutons la ligne suivante :

JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64

Qui est la variable d'environnement pour que tomcat9 soit fonctionnelle.

- Nours devons configurer la tomcat9 manager afin que tomcat9 soit accessible. Pour cela, nous ajoutons les deux lignes suivantes :

role rolename=”admin-gui”/
user username=”admin” password=”password” roles=”manager-gui,admin-gui””/

Dans le fichier **" tomcat-users.xml "** qui se trouve dans le dossier **" /etc/tomcat9 "** .

- A ce stade, tomcat9 est fonctionnelle, pour le démarrer :

systemctl start tomcat9

- Pour l'arrêter :

systemctl stop tomcat9

- Après avoir installé Tomcat9 et Maven, nous allons procéder à l'installation d'Apereo-CAS. Pour cela, nous commençons par installer le projet nécessaire à l'utilisation de cas-overlay-template

-Dans le dossier **/opt** , nous allons utiliser la commande :

git clone https ://github.com/apereo/cas-overlay-template

- Dans le fichier **build.gradle** qui se trouve dans le dossier **cd /opt/cas-overlay-template**, nous allons ajouter les trois lignes dans la section dépendances puis enregistrer le fichier :

```
dependencies {
implementation "org.apereo.cas :cas-server-webapp-init : $ {casServerVersion}”
implementation "org.apereo.cas :cas-server-support-ldap : $ {project.’cas.version’}”
implementation"org.apereo.cas :cas-server-support- json-
servregistry :${casServerVersion}”
}
```

-Nous ajoutons dans le fichier **cas.properties** la configuration de LDAP :

```
cas.server.name=http ://192.168.36.108 :8080
cas.server.prefix=${cas.server.name}/cas
logging.config : file :/etc/cas/config/log4j2.xml
```

```
cas.authn.accept.users=
cas.authn.ldap\[0\].providerClass=org.ldaptive.provider.unboundid.UnboundIDProvider
cas.authn.ldap\[0\].type=AUTHENTICATED
cas.authn.ldap\[0\].useSsl=false
cas.authn.ldap\[0\].ldapUrl=ldap ://domain.univ :389
cas.authn.ldap\[0\].baseDn= dc= domain,dc=univ
cas.authn.ldap\[0\].subtreeSearch=true
cas.authn.ldap\[0\].searchFilter=sAMAccountName=\{user\}
cas.authn.ldap\[0\].principalAttributeList=cn,givenName,mail
cas.authn.ldap\[0\].bindDn=CN=Admincas,CN=CasAdmin,DC= domain,DC=univ
cas.authn.ldap\[0\].bindCredential=P@ssW0rd
```

- A présent, nous vérifions le port de connexion LDAP 389 avec la commande :

telnet 192.168.36.108 389

- Nous avons eu ce résultat :

Trying 192.168.36.108...
Connected to 192.168.36.108.
Escape character is ˆ]

Etape 2 : Installation de Gradle

Gradle est un outil d'automatisation de construction open source conçu pour être suffisamment flexible pour créer presque tous types de logiciels [34]. Nous allons à présent commencer par la commande **./gradlew clean** qui va récupérer les dépendances pour notre projet. Ensuite, les deux commandes : **./gradlew clean copyCasConfiguration build** et **./gradlew createKeystore** respectivement [35].

-Une fois l'installation de Gradle est faite, nous allons copier le fichier **cas.war** et le mettre dans l'arborescence **webapps** de Tomcat9 avec la commande suivante :

cp /opt/cas-overlay-template/build/libs/cas.war /var/lib/tomcat9/webapps/

-Nous relançons le service de tomcat9 comme suit :

systemctl restart tomcat9.service

-Maintenant, nous allons tester la connexion. Ici notre adresse IP est :

http ://192.168.36.108 :8080/cas
-La page d'authentification CAS apparait comme la montre la figure (III.6) :

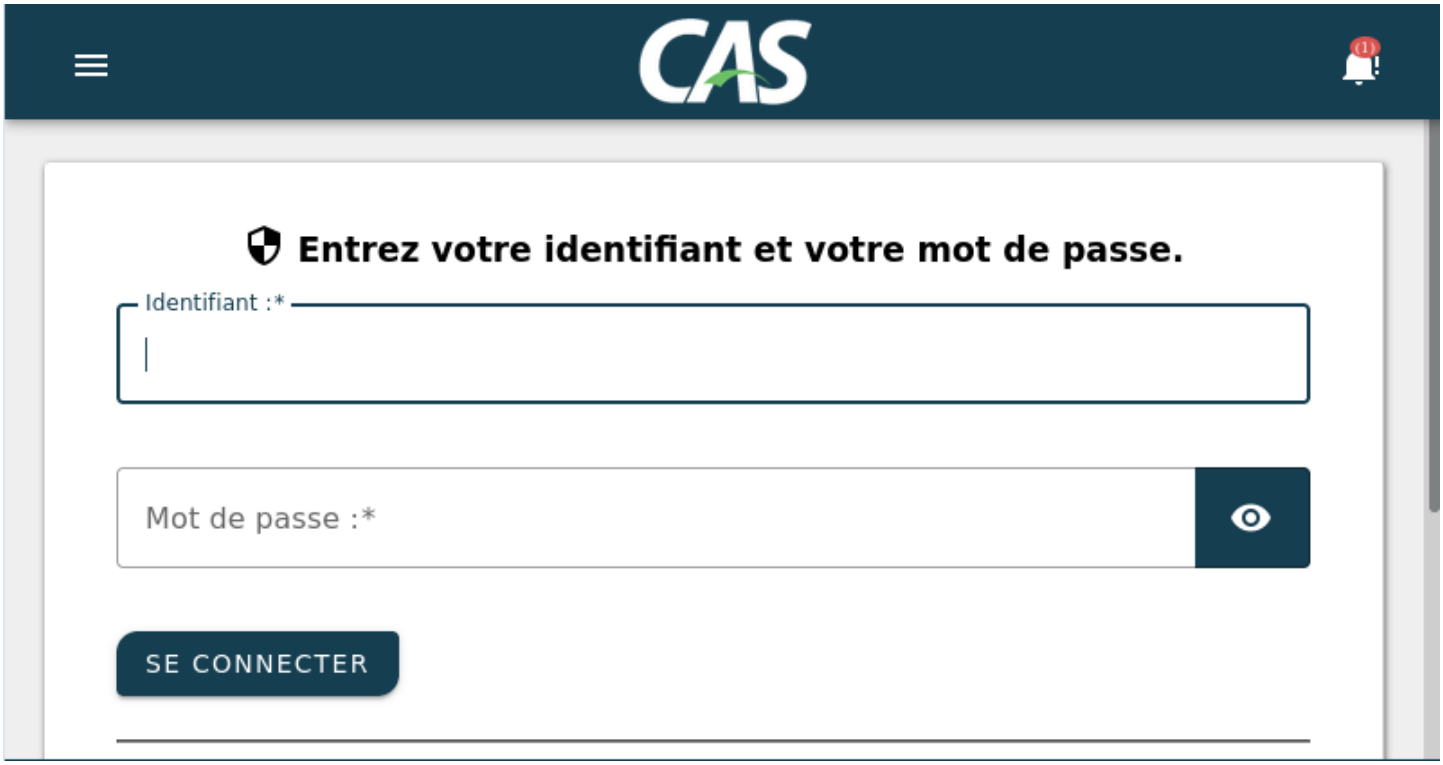

Figure III.6 – Interface d'authentification de CAS.

III.4 Réalisation de l'interface d'authentification avec le service Java-RMI méthodes

A. Présentation de Java-RMI :

RMI (Remote Method Invocation) est une API JAVA permettant l'appel, l'exécution et le renvoi du résultat d'une méthode exécutée dans une machine virtuelle différente de celle de l'objet l'appelant. La machine sur laquelle s'exécute la méthode distante est appelée serveur [36].

B. Outils de développement de l'application :

Nous avons utilisé NetBeans qui est un environnement de développement intégré (IDE) pour Java pour notre application. La figure(III.7 représente l'interface de notre application :

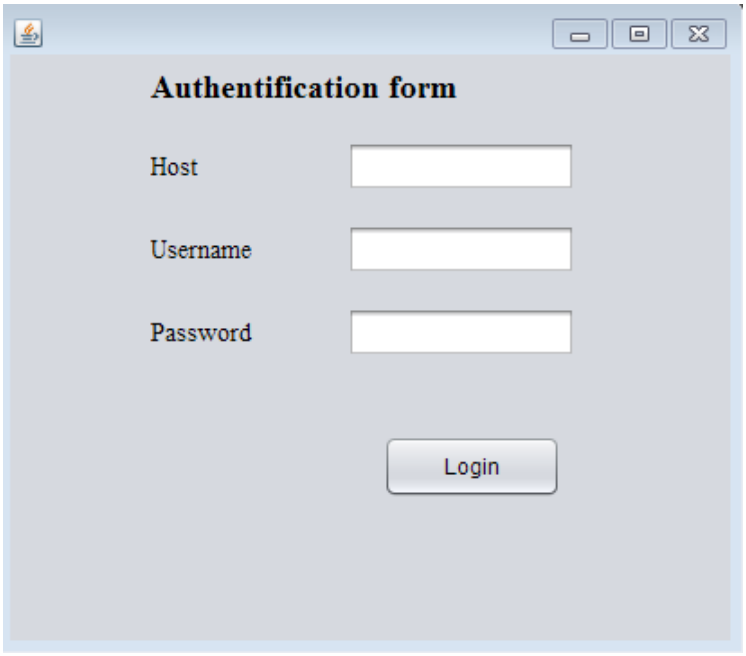

Figure III.7 – Interface de l'application.

III.5 Conclusion

Dans ce chapitre, nous avons élaboré les différentes étapes suivi afin de mettre en œuvre notre projet, nous avons procéder en premier, par présenter notre projet. Par la suite, nous avons expliqué les diverses phases de l'installation et configuration du serveur CAS et l'annuaire LDAP. De plus, nous avons défini le service web sur lequel l'interface client fonctionne.

Conclusion générale

Pour conclure, actuellement, le monde connait des avancées très significatives dans différents domaines de l'informatique, notamment dans la sécurité et l'authentification.

L'objectif du présent travail est la mise en place d'un système d'authentification unique pour certaines applications web liées, mais indépendantes, sans que leur utilisateur soit invité, au cours d'une session particulière, à se reconnecter à chacune d'entre elles. En ce sens et d'un point de vue technique, le Single-Sign-On permet de centraliser les systèmes d'authentification et d'accorder aux utilisateurs une meilleure gestion de leurs mots de passe. De plus, avec le Single-Sign-On, la sécurisation des comptes se fait à tous les niveaux d'entrée, de sortie et même au niveau de l'accès aux systèmes, sans que l'utilisateur soit toujours sollicité. Raisons pour laquelle, nous avons largement exploité le SSO. Cependant, bien que le Single-Sign-On soit pratique, il a tout de même des inconvénients. En effet, si un compte SSO est piraté, cela peut entrainer une panne, et conduit à la perte des données des autres comptes qui sont sous le même système d'authentification.

Outre le SSO, afin d'atteindre l'objectif visé dans ce projet, nous avons opté pour une méthode et des outils bien décrits dans la partie théorique, spécifiquement l'annuaire LDAP et le serveur d'authentification CAS. L'annuaire LDAP est l'objet de stockage et d'enregistrement des informations concernant les utilisateurs, tandis que le serveur d'authentification CAS est utilisé pour implémenter le système d'authentification unique (SSO).

Par ailleurs, au cours de la réalisation de ce projet, nous avons acquis de nouvelles connaissances pratiques, notamment, en ce qui concerne la configuration du DNS (Domain Name Server), la configuration de l'annuaire LDAP ainsi que celles de Central Authentification Service. Néanmoins, nous avons aussi rencontré plusieurs problèmes notamment le manque de documentation sur le serveur Central Authentification Server ainsi que l'interdépendance entre les versions des outils installés et configurés.

Enfin, comme perspective à la fin de ce projet, nous proposons une meilleure façon de réduire les risques, soit la combinaison entre le Single-Sign-On et l'authentification multifacteurs (MFA). Cette dernière est une méthode d'authentification dans laquelle l'utilisateur doit fournir au minimum deux facteurs de vérification pour accéder à une ressource de type application. La combinaison SSO et MFA améliore la sécurité des comptes des utilisateurs et rend le SSO plus fiable que lorsqu'il est utilisé seul.

Annexes

Kerberos

Définition

Kerberos est un protocole d'authentification réseau. Il est conçu pour fournir une authentification forte pour les applications client/serveur en utilisant la cryptographie à clé secrète. Une implémentation gratuite de ce protocole est disponible auprès du MIT (Massachusetts Institute of Technology). Kerberos est également disponible dans de nombreux produits commerciaux [37].

Le système Kerberos se compose d'un serveur d'authentification AS (Authentification Service), qui permet à l'utilisateur de s'authentifier une seule fois pendant la durée d'une session ; d'un serveur de tickets TGS (Ticket Granting Service), qui génère le ticket de service demandé par l'utilisateur pour se connecter au service demandé. Et enfin d'un centre de distribution de clés, qui assure la liaison entre les deux serveurs AS et TGS [38].

Fonctionnement [39]

1. L'utilisateur demande un ticket d'authentification Ticket Granting Ticket, (TGT) au Key Distribution Center (KDC).

2. Le KDC vérifie les données d'identification et renvoie un TGT chiffré et une clé de session.

3. Le TGT est chiffré à l'aide de la clé secrète TGS.

4. L'utilisateur conserve le TGT et, lorsqu'il arrive à expiration, le gestionnaire de session local en demande un autre.

Si l'utilisateur demande à accéder à un service ou à une autre ressource du réseau, la procédure est la suivante :

1. L'utilisateur envoie le TGT encours du TGS avec le Service Principal Name (SPN) de la ressource à laquelle l'utilisateur souhaite accéder.

2. Le KDC vérifie le TGT de l'utilisateur et s'assure que l'utilisateur a accès au service.

3. Le TGS envoie une clé de session valide pour le service à l'utilisateur.

4. L'utilisateur transmet la clé de session au service pour prouver que l'utilisateur dispos d'un accès, et le service accorde l'accès.

Avantages [38] :

— Transmission des mots de passe cryptés à travers le réseau ;

— Un espion sur le réseau ne doit pas pouvoir obtenir l'information nécessaire pour se faire passer pour un utilisateur ;

— Kerberos doit pouvoir se reposer sur une architecture de serveur distribuée avec des systèmes interchangeable.

Inconvénients [40] :

— Kerberos a des délais stricts ; les horloges des hôtes concentrés doivent être synchronisées avec l'horloge du serveur Kerberos pour s'assurer que l'authentification n'échoue pas ;

— Kerberos a besoin que le serveur central soit disponible en continu. Lorsque le serveur Kerberos test en panne, personne ne peut ouvrir une session ;

— Comme toute l'authentification est contrôlée par un KDC centralisé, tout compromis dans cette infrastructure, tel que le mot de passe de l'utilisateur pour un poste de travail local volé, peut permettre à un attaquant de se faire passer pour n'importe quel utilisateur.

Bibliographie

[1] https://www.esup-portail.org/consortium/espace/SSO_1B/cas/jres/cas-jres2003-article-web.htm, (consulté le 13 Juin 2022).

[2] NARCISSE K and ERIC M. " mise en œuvre d'un système d'authentification centralisé sso avec fournisseur d'identités ". Mémoire de licence, Université de Dschang/iut-fv de Bandjoun, 2011/2012.

[3] RAPHAEL Y. " support de cours de sécurité informatique et crypto ". Mémoire de master, université de Congo-Kinshasa, 2018.

[4] RHARRAB A. " audit sécurité des systèmes d'information ". Mémoire de projet de fin d'études, université Mohammed V Agdal Rabat Maroc, 2011/2012.

[5] http://www.mcours.com/telecharger/la-securite-informatique-lutter-contre-les-attaques-et-le-hacking/, (consulté le 10 Avril 2022).

[6] https ://schaellerclement.wordpress.com/veille-technologique., (consulté le 2 Avril 2022).

[7] http://www.wikayanet.dz/index.php/fr/dossiers-securite/1178-la-securite-parlons-en-serieusement, (consulté le 2 Avril 2022).

[8] https://waytolearnx.com/2018/07/difference-entre-attaque-active-et-attaque-passive.html, (consulté le 19 Avril 2022).

[9] https://www.securiteinfo.com/attaques/hacking/typesattaques.shtml, (consulté le 2 Avril 2022).

[10] MAHAMMEDI N and MAHDADI H. " implémentation de bechemark d'opération crypto basées ecc pour l'étude et comparaison de courbes elliptiques fp et f2n ". Mémoire de master, université de KASDI MERBAH OUARGLA, 2013.

[11] DJAFRI N and DJAFRI H. " implémentation d'un protocole d'authentification et de partage de clés dans un système distribué ". Mémoire de fin de cycle, université de Béjaia, 2015/2016.

[12] MIHOUBI M and MEDJANI N. " sécurisation d'une infrastructure lan/wan à base d'équipement cisco ". Mémoire de master, université de Mouloud Mammeri TIZI-OUZOU, 2015.

[13] https://www.futura-sciences.com/tech/definitions/informatique-antivirus-10999/, (consulté le 25 Avril 2022).

[14] HELAL S. " authentification anonyme et contrôle d'accès dans un environnement cloud : Application au domaine e-santé ". Mémoire de master en informatique, université Saad Dahlab Blida, 2018/2019.

[15] DWITI P, KHUSHBOO R, SNEHA T, and TANI N. " an overview of various authentication methods and protocols ". 131(9) :25–27, 2015.

[16] KHERBACHE M and LETAT Z. " proposition et implémentation d'un protocole d'authentification unique ". Mémoire de fin de cycle, université de BEJAIA, 2015/2016.

[17] https://www.oracle.com/fr/security/qu-est-ce-qu-un-sso.html, (consulté le 23 Avril 2022).

[18] http://www-igm.univ-mlv.fr/~dr/XPOSE2006/CLERET/objectifs.html#oneUser, (consulté le 2 Mai 2022).

[19] www.commentcamarche.com/livreblancsurSSO, (consulté le 5 Avril 2022).

[20] http://anthonyreault.free.fr, (consulté le 5 Mai 2022).

[21] BENGLIA A and BENHAMOUDA A. " authentification sso des services web du campus universitaire d'ouargla ". Mémoire de master, université d'OUARGLA, 2015/2016.

[22] http://monge.univ-mlv.fr/~dr/XPOSE2009/Sign%20Sign%20On/archi.html, (consulté le 6 Mai 2022).

[23] www.cert-ist.com, (consulté le 10 Mai 2022).

[24] MIHI A and TERBAGOU A. " authentification unifiée pour l'accès aux services web de l'université ". Mémoire de master, université Kasdi Merbah Ouargla, 2012/2013.

[25] ADJAOUD Y and KEHOUL T. " authentification unique avec cas et ldap ". Mémoire de master, université de Bejaia, 2011/2012.

[26] https://www.institut-numÃ]rique.org/22-conception-51ee914757753/amp, (consulté le 13 Juin 2022).

[27] https://www.purdue.edu/securepurdue/identity-access/boilerkey/CAS-information.php, (consulté le 21 Juin 2022).

[28] https://www.varonis.com/fr/blog/difference-entre-active-directory-et-ldap, (consulté le 21 Juin 2022).

[29] PLOUIN G, SOYER J, and TRIOULLIER M-E. " sécurité des architectures web ". Ouvrage, DUNOD, 2004.

[30] http://idum.fr/spip.php?article326, (consulté le 10 Juin 2022).

[31] http://koretelecoms.over-blog.com/2020/10/comment-creer-des-comptes-unix-sur-l-annuair
html, (consulté le 19 Juin 2022).

[32] https://www.editions-eni.fr/open/mediabook.aspx?idR=98db25384380c51b79e46e170641bf45,
(consulté le 19 Juin 2022).

[33] https://www.jmdoudoux.fr/java/dej/chap-maven.htm, (consulté le 10 Juin 2022).

[34] https://docs.gradle.org/current/userguide/what_is_gradle.html, (consulté le 21
Juin 2022).

[35] https://www.osnetworking.com/fr/apereo-cas/, (consulté le 21 Juin 2022).

[36] https://www.jmdoudoux.fr/java/dej/chap-rmi.htm, (consulté le 10 Juin 2022).

[37] www.web.mit.edu/Kerberos/, (consulté le 18 Juin 2022).

[38] KHERBACHE M and LETAT Z. " proposition et implémentation d'un protocole d'authentification unique ". Mémoire de fin de cycle, université de Béjaia, 2015/2016.

[39] https://www.varonis.com/fr/blog/explication-de-lâĂŹauthentification-kerberos,
(consulté le 21 Juin 2022).

[40] https://docs.citrix.com/fr-fr/citrix-adc/current-release/aaa-tm/
configuring-commonly-used-protocols/citrix-adc-aaa-with-kerberos-ntlm.html,
(consulté le 21 Juin 2022).

Résumé

La saisie d'identifiants différents pour chaque application entraîne des interruptions de parcours. Une solution SSO conviviale et sécurisée peut protéger vos applications et vos utilisateurs, tout en améliorant la productivité et la commodité.

Le présent travail est reparti en deux parties. La première est consacré à faire un rappel sur les fondamentaux de l'authentification et la sécurité informatique .Notre objectif est de donner un aperçu global de ce qu'est l'authentification unique ou le SSO, le serveur CAS ainsi que l'annuaire LDAP sur le point théorique. La deuxième est une mise en application d'un système d'authentification unique pour certaines applications web.

Mots clés : Authentification, Single-Sign On, Central Authentification Service, LDAP.

Abstract

Entering different identifiers for each application leads to interruptions in the journey. A user-friendly and secure SSO solution can protect your applications and users, while improving productivity and convenience.

This work is divided into two parts. The first is devoted to recalling the fundamentals of authentication and computer security. Our objective is to give an overview of what single sign-on or SSO is the CAS server and the LDAP directory. On a theoretical point . The second is an implementation of a single sign-on system for certain web applications.

Keys words : Authentification, Single-Sign On, Central Authentification Service, LDAP.