Institut Universitaire de Technologie,
Aix-Marseille Université

RAPPORT DE STAGE de fin de deuxième année
Bachelor Universitaire de Technologie
Spécialité Réseaux et Télécommunications
parcours cybersécurité

Mise en place de la solution client-serveur FreeIPA

Antoine Planas

THALES

Responsable entreprise : Franck Morier

Responsable académique : Arnaud Février

2023

Table des matières

1 Introduction

En 2019, l'entreprise Gemalto, spécialiste de la sécurité numérique, se fait racheter par le Groupe Thalès. La filiale Thalès DIS (Digital Identity and Security), située à Meyreuil, est une ancienne filiale de l'entreprise Gemalto. Dans le cadre de ce rachat, ainsi que pour des questions des sécurités des données, une migration du réseau interne doit être réalisée vers le réseau du Groupe.

C'est au cours de cette situation que j'ai eu l'occasion d'intervenir, au sein de l'entreprise Thalès, en tant que stagiaire, me permettant également de valider ma deuxième année de BUT, Bachelor Universitaire de Technologie, en Réseaux et Télécommunications. Je fus placé dans le service IT, Information Technology, me permettant d'être au plus proche du réseau mais également de découvrir plus profondément, l'administration des systèmes.

Un objectif m'a initialement été confié à mon arrivé dans l'entreprise, qui été de mettre en place une solution de backup du réseau, mais qui, par faute de non-réception du matériel nécessaire, n'a pas pu être abordée. Une autre mission m'a alors été assignée : le déploiement de la solution client-serveur FreeIPA.

Au cours de ce rapport, je vais vous présenter les différentes tâches que j'ai eu à réaliser au cours de mon stage ainsi que la démarche que j'ai suivi pour l'accomplissement de ma mission. Dans un premier temps, je vous présenterai la solution FreeIPA et les besoins de l'entreprise, puis, dans un second temps, je vous expliquerai la démarche que j'ai suivi pour le déploiement de cette solution.

2 Présentation de FreeIPA et des besoins

Comme énoncé dans l'introduction, lors de ce stage, j'ai eu pour objectif de mettre en place la solution FreeIPA sur le réseau de l'entreprise. Ne connaissant pas ou n'étant pas familier à certains concepts, j'ai dû commencer par me renseigner et apprendre à maîtriser ces nouveaux concepts.

2.1 Présentation de FreeIPA

N'ayant aucune piste mais uniquement l'objectif final en tête, j'ai alors commencé mes recherches en me renseignant sur FreeIPA. Ce point de départ dans mes recherches me permettra, au fur et à mesure de celles-ci, de découvrir de nouvelles notions et de m'offrir de nouvelles pistes de recherches.

FreeIPA, IPA pour Identity, Policy et Audit est une solution intégrée de gestion des identités et des accès open-source. Il combine plusieurs composants, notamment LDAP (Lightweight Directory Access Protocol), Kerberos, DNS (Domain Name Server) et Dogtag pour les certificats, pour fournir un ensemble complet d'outils permettant de gérer les identités, les authentifications et les autorisations dans un environnement informatique. FreeIPA simplifie la gestion des utilisateurs, des groupes, des politiques de sécurité et des services dans une infrastructure.

2.1.1 Fonctionnalités de FreeIPA

Due à son architecture, composée d'un ensemble de plusieurs services, FreeIPA offre énormément de fonctionnalités de gestion et de sécurité de notre environnement informatique qui sont souvent essentielles pour un administrateur. Via son composant LDAP (voir annexe 1), FreeIPA permet une gestion centralisée des identités. Sans ce type de service, il est difficile de gérer la répartition des utilisateurs, des groupes et des rôles d'une entreprise de plusieurs dizaines d'employés. LDAP permet de regrouper et administrer ces informations dans des annuaires.

Aujourd'hui, les entreprises utilisent de plus en plus de services différents. Pour garantir la sécurité des données et des utilisateurs, la plupart des services nécessitent désormais de créer un compte avec un identifiant et un mot de passe. Pour une sécurité optimale, un utilisateur devrait utiliser un mot de passe complexe et différent pour chacun des services qu'il utilise. Il serait difficile et fastidieux de retenir plusieurs dizaines de mots de passe et de devoir se connecter à chaque utilisation d'un service. Grâce à Kerberos (voir annexe 2), FreeIPA permet à l'utilisateur de s'authentifier une seule fois pour utiliser tous ses services. Avec cette authentification unique, SSO (Single Sign-On, voir annexe 3), l'utilisateur n'a plus besoin de réintroduire ses informations d'authentification à chaque utilisation de ses services.

Pour une entreprise, la sécurité des données ainsi que celle des utilisateurs est un aspect vraiment important de l'informatique. Pour Thalès, dont certains collaborateurs travaillent sur des projets extrêmement confidentiels, une protection forte, physique et informatique est nécessaire. Avec FreeIPA, il est possible de créer des règles de protection. Par exemple, il est possible de créer une règle permettant l'accès à un hôte uniquement à certains collaborateurs. Les autres utilisateurs, n'ayant pas les permissions d'accès, ne pourront donc pas se connecter sur cet hôte. Il est également possible de contrôler d'autres aspects de sécurité tel que les mots de passe ou l'utilisation des commandes administrateur.

FreeIPA intègre nativement énormément de fonctionnalités et il serait trop long de toutes les énumérer. De plus, FreeIPA offre la possibilité d'ajouter son propre service, ce qui peut être utile si celui-ci n'est pas intégré par défaut dans la solution. Cette fonctionnalité permet alors d'accroître encore le nombre des possibilités qu'offre la solution.

2.1.2 Utilisations de FreeIPA

Comme expliqué précédemment, FreeIPA est une solution complète offrant énormément de possibilités. Toutes ces fonctionnalités, disponibles grâce à son architecture, fait de FreeIPA une solution polyvalente, pouvant trouver son utilité dans une grande variété d'environnement. Aujourd'hui la plupart des environnements, qu'ils soient personnels ou appartenant à une entreprise, ont besoins d'avoir accès à certains services pour le bien-être et la protection des utilisateurs.

FreeIPA est énormément répandu dans les environnements Linux. Sur les environnements Linux, la solution est facile d'accès et sa mise en place est relativement simple et rapide, quelques soit l'OS (Operating System, système d'exploitation en français). Les documentations de la communauté Linux pour la mise en place de cet outil sont nombreuses, ce qui permet une résolution aisée des erreurs. FreeIPA est également disponible sur MacOS mais les documentations sont moins nombreuses et sa mise en place est plus complexe. La solution n'est cependant pas disponible sur le système d'exploitation de Microsoft. En effet, Windows n'intègre pas la mise en place de FreeIPA car Microsoft délivre ses propres services tel que Microsoft Active Directory (AD, voir annexe 4). Il est tout de même possible, avec quelques ajustements, d'installer FreeIPA sur des machines Windows.

Généralement, beaucoup d'entreprises ou d'organisations reposent sur des systèmes Linux. Celles-ci doivent alors mettre en place un grand nombre de services correspondant à leurs besoins, afin de garantir la sécurité de leur environnement. FreeIPA, étant une solution complète composée de plusieurs services, est alors un choix privilégié pour les entreprises. Sa variété de services nativement disponible mais également sa flexibilité dans l'ajout de nouveaux services permet aux entreprises d'avoir un outil répondant à tous leurs besoins.

Le fait que FreeIPA soit une solution open-source ne limite pas son accessibilité aux entreprises. Il est tout à fait possible d'installer les paquets et de mettre en place un serveur et des clients FreeIPA sur un réseau domestique. Un utilisateur pourrait alors déployer la solution sur son réseau privé et ainsi sécuriser et contrôler les accès à ses clients mais également gérer les différents utilisateurs ainsi que leurs permissions.

De plus, comme FreeIPA est un projet open-source, il est possible, pour les amateurs de développement, de créer de nouvelles fonctionnalités permettant une meilleure expérience. Par défaut, il n'y a pas de fonctionnalité pour afficher une photo dans le profil d'un utilisateur, mais il existe une extension permettant de mettre en place cette fonctionnalité.

En résumé, FreeIPA offre une multitude de fonctionnalités qui simplifient la gestion des utilisateurs, des groupes, des politiques de sécurité et des services au sein d'une infrastructure informatique. Son architecture combinant différents composants tels que LDAP, Kerberos et DNS, permet une gestion centralisée et sécurisée des identités. Les entreprises bénéficient de ses capacités avancées pour garantir la protection des données et des utilisateurs, ainsi que pour contrôler les accès aux ressources sensibles. De plus, FreeIPA est largement adopté dans les environnements Linux, offrant une mise en place facilitée et une compatibilité étendue. Son caractère open-source favorise également la flexibilité et l'ajout de nouvelles fonctionnalités selon les besoins spécifiques. Enfin, même dans un contexte personnel, l'installation de FreeIPA sur un réseau domestique permet de renforcer la sécurité et de gérer efficacement les accès et les utilisateurs. FreeIPA se positionne donc comme une solution polyvalente et puissante pour répondre aux exigences croissantes de gestion des identités et des accès, offrant des avantages significatifs aux entreprises et aux utilisateurs.

2.2 Analyse des besoins

Après mettre renseigner sur cette solution FreeIPA, j'ai eu la nécessité de connaître les besoins de l'entreprise. Tous les environnements et toutes les situations ne sont pas identiques. Chaque entreprise ou utilisateur à ses propres besoins et ses propres contraintes. L'analyse des besoins est donc une étape essentielle et non négligeable lors de la mise en place de telle solution. De plus, il est également parfois nécessaire de se renseigner sur les besoins en ressources matérielles des solutions et outils que nous souhaitons installer pour leur bon fonctionnement.

2.2.1 Besoins et contraintes de l'entreprise

Comme j'ai pu le stipuler préalablement, j'ai la nécessité de me renseigner sur les besoins de l'entreprise. Cette compétence d'analyse des besoins n'est pas unique au domaine informatique. Quel que soit le domaine, analyser les besoins d'un client, de l'entreprise, de la solution ou même ses propres besoins, est une étape indispensable. Par exemple, un agent immobilier doit se renseigner sur les besoins de ses clients afin de trouver un logement leur correspondant.

À l'instar d'un agent immobilier, j'ai discuté avec l'équipe IT afin de connaître nos véritables besoins, mais également nos contraintes. Comme énoncé dans l'introduction, le réseau interne de l'agence Thalès de Meyreuil est actuellement en cours de migration vers le réseau Thalès. De plus, le réseau interne du site est lui-même composé de deux réseaux : un réseau « blanc », utilisé par les collaborateurs pour des projets à faible confidentialité, et un réseau « rouge », isolé et utilisé pour des projets à forte confidentialité.

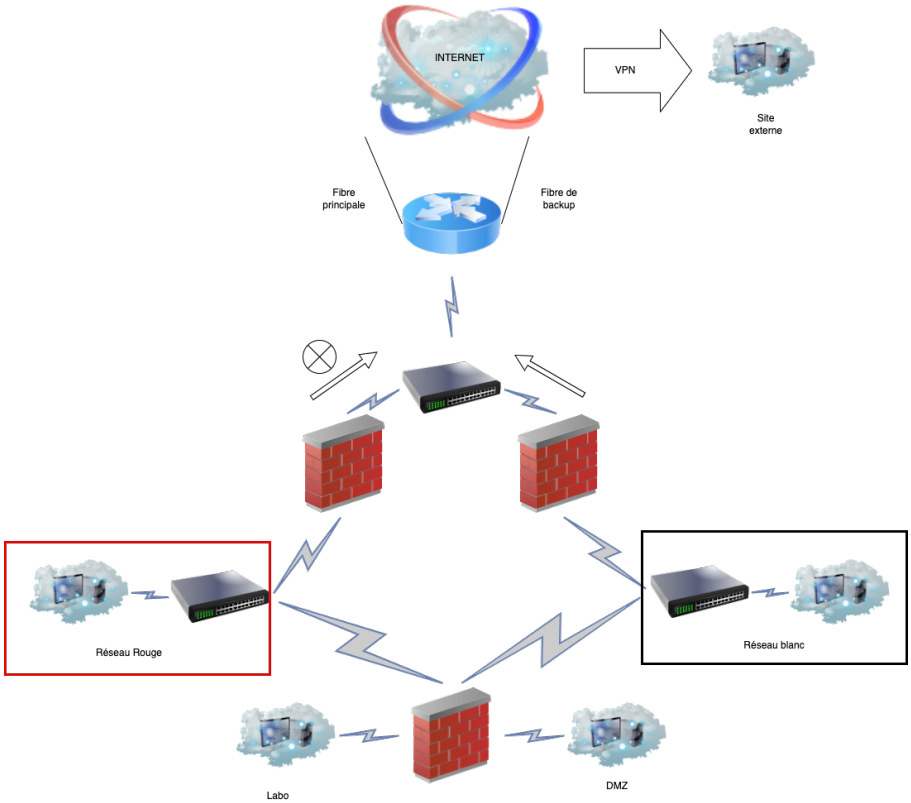

Image 1 : Schémas simplifiés du réseau

L'individualité de l'architecture de ce réseau apporte alors des contraintes ainsi que des besoins de service et de protection. La confidentialité de certains projets nécessite une sécurité adéquate pour accéder à des ressources sensibles. Sans certaines mesures de sécurité, il serait tout à fait possible d'accéder à de tels données, qu'on soit un collaborateur de l'entreprise ou non. Pour garantir cette

confidentialité, il m'est nécessaire de créer des règles et donner des permissions d'accès uniquement aux personnes qui ont le droit et le besoin d'y accéder.

Par exemple, on peut imaginer une entreprise composée de trois services (IT, Analog et Digital) et possédant deux clients (A et B). Il se pourrait que pour des raisons de confidentialité, comme dans mon cas, uniquement les services IT et Analog ont le droit d'accéder au client A et uniquement les services IT et Digital peuvent accéder au client B. Il me faut alors, après avoir déterminé ces droits, créer des règles d'accès pour chaque client, permettant ainsi de répondre aux besoins de mon entreprise et de ses collaborateurs.

Cependant, pour protéger la confidentialité et garantir une sécurité optimale, se reposer uniquement sur des règles permettant de contrôler les accès n'est pas suffisant. Pour renforcer la protection du réseau et de ses ressources, il faut mettre en place d'autres sécurités contrôlant ces accès. Il m'est alors nécessaire de contrôler également les méthodes d'accès. Par exemple, si pour des réseaux de confidentialité, mon utilisateur n'a pas le droit de se connecter à distance sur mon client, il faut que je puisse contrôler et empêcher les méthodes d'accès sur mon client.

Le contrôle d'accès est un moyen véritablement efficace pour améliorer la sécurité d'un réseau informatique et pour garantir la sécurité de données mais n'est pas suffisant. De nos jours, l'authentification avec un identifiant et un mot de passe est énormément répandue. Cependant, les utilisateurs ne se rendent pas toujours compte de l'importance de ces mots de passe. La quantité de services nécessitant ce type d'authentification étant de plus en plus nombreux, nous réutilisons souvent un même mot de passe ou un mot de passe similaire d'un service à l'autre. Certaines personnes pensent à trouver un nouveau mot de passe pour chaque service mais négligeant ainsi la robustesse de celui-ci, afin de pouvoir se souvenir de dizaines de mots de passe. Certes, il existe désormais des applications ou des services qui offrent la possibilité de créer et gérer des mots de passe robustes, mais cela ne garantit pas toujours leur protection. Il est nécessaire de modifier régulièrement son mot de passe afin de garantir une sécurité optimale.

Afin de protéger le réseau ainsi que les utilisateurs, je me dois alors créer des règles permettant de gérer les mots de passe des différents utilisateurs, leur imposant l'utilisation de mots de passe robustes, le changement régulier et l'utilisation d'un mot de passe qui n'a pas déjà été utilisé. Cela permettra de rendre la découverte d'un mot de passe par un pirate plus complexe, voire quasiment impossible en additionnant ces différentes conditions.

COMBIEN DE TEMPS FAUT-IL À UN PIRATE POUR TROUVER VOTRE MOT DE PASSE EN 2023 ?

Nombre de caractères	Nombres seulement	Lettres minuscules	Lettres majuscules et minuscules	Nombres, lettres majuscules et minuscules	Nombres, lettres majuscules et minuscules, symboles
4	Immédiat	Immédiat	Immédiat	Immédiat	Immédiat
5	Immédiat	Immédiat	Immédiat	Immédiat	Immédiat
6	Immédiat	Immédiat	Immédiat	Immédiat	Immédiat
7	Immédiat	Immédiat	1 seconde	2 secondes	4 secondes
8	Immédiat	Immédiat	28 secondes	2 minutes	5 minutes
9	Immédiat	3 secondes	24 minutes	2 heures	6 heures
10	Immédiat	1 minute	21 heures	5 jours	2 semaines
11	Immédiat	32 minutes	1 mois	10 mois	3 ans
12	1 seconde	14 heures	6 ans	53 ans	226 ans
13	5 secondes	2 semaines	332 années	3 000 années	15 000 ans
14	52 secondes	1 an	17 000 ans	202 000 ans	1 million d'années
15	9 minutes	27 ans	898 000 ans	12 millions d'années	77 millions d'années
16	1 heure	713 ans	46 millions d'années	779 millions d'années	5 milliards d'années
17	14 heures	18 000 ans	2 milliards d'années	48 milliards d'années	380 milliards d'années
18	6 jours	481 000 ans	126 milliards d'années	1 trillion d'années	26 trillions d'années

Infographie FRANCENUM.GOUV.FR Source : Hive Systems

Image 2 : La robustesse des mots de passe

2.2.2 Les conteneurs

A l'instar de nombreuses autres entreprises, chez Thalès, nous utilisons de la virtualisation de machines afin de faciliter l'administration du réseau mais également pour réduire les coûts financiers. En effet, virtualiser des machines permet de réduire leur nombre et par conséquent, réduire les coûts d'acquisitions et de maintenance du matériel, sans oublier les coûts électriques.

La virtualisation est une technologie qui permet de créer plusieurs environnements informatiques isolés, appelés machines virtuelles (VM), sur une seule machine physique. Chaque VM dispose de son propre système d'exploitation, de ses ressources dédiées et peut exécuter des applications de manière indépendante. Cela permet d'optimiser l'utilisation des ressources matérielles et d'isoler les applications les unes des autres.

Les conteneurs sont une autre forme de virtualisation, mais avec une approche plus légère et plus portable. Un conteneur est une unité d'exécution autonome qui contient tout ce dont une application a besoin pour s'exécuter. Contrairement aux machines virtuelles, les conteneurs partagent le même noyau d'exploitation et les bibliothèques du système hôte, ce qui les rend plus légers et plus rapides à démarrer.

La différence clé entre les conteneurs et les machines virtuelles réside dans leur niveau d'isolation et d'abstraction. Les machines virtuelles offrent une isolation complète, où chaque VM fonctionne comme une entité autonome avec son propre système d'exploitation et ses ressources dédiées. En revanche, les conteneurs partagent le système d'exploitation de l'hôte, ce qui leur permet d'être plus légers et d'offrir une meilleure performance. Ils isolent les applications au niveau des processus, des fichiers et des réseaux, mais utilisent les ressources du système d'exploitation hôte pour leur exécution.

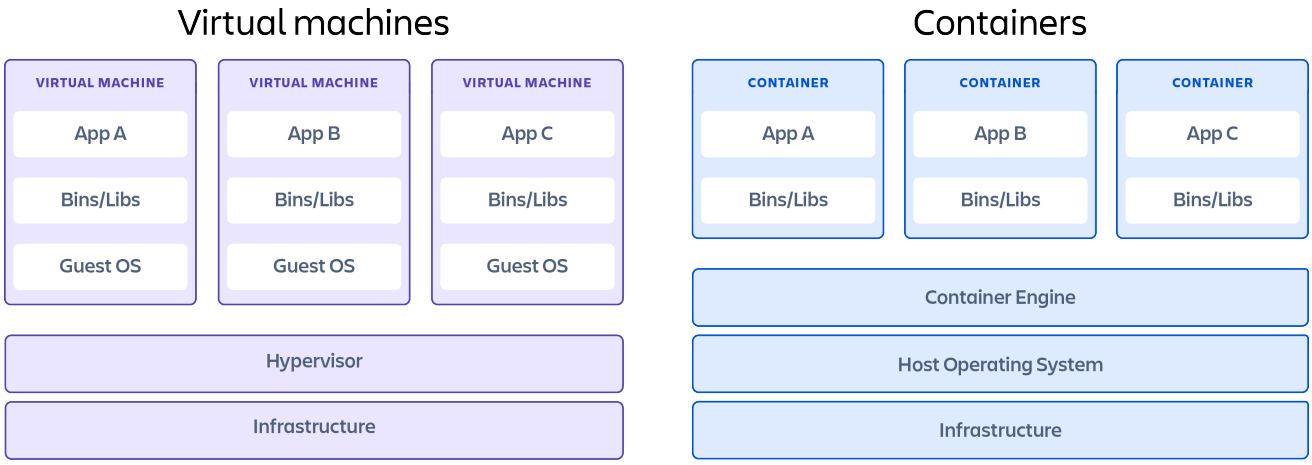

Image 3 : Différences d'isolation entre une VM et un conteneur

Docker et Podman sont des outils populaires utilisés pour la gestion des conteneurs. Docker est une plateforme de conteneurisation qui simplifie le processus de création, de distribution et de déploiement des applications dans des conteneurs. Il fournit un environnement de développement cohérent et portable, permettant aux développeurs de créer des conteneurs facilement.

D'un autre côté, Podman est un autre outil de gestion de conteneurs qui offre une alternative à Docker, supporté par RedHat. Il permet de créer et de gérer des conteneurs de manière similaire, mais avec une approche plus orientée vers les conteneurs de type "sandbox". Podman utilise le même format d'image que Docker, ce qui facilite la transition entre les deux outils.

Au cours de mon cursus, j'ai eu l'occasion d'apprendre à maitriser les bases de la virtualisation sur des machines virtuelles mais, même si je l'ai abordé rapidement en cours, je n'avais aucune connaissance

sur la virtualisation de conteneurs. Chez Thalès, la virtualisation se fait principalement avec des conteneurs et notamment avec des conteneurs Podman. L'essentiel des machines fonctionnent avec un OS RedHat, il était alors logique d'utiliser Podman. Il m'a donc fallu apprendre à maîtriser la virtualisation de conteneurs en commençant par l'apprentissage de Docker puis celui de Podman.

En résumé, cette étape d'analyse indispensable m'a permis de connaitre les besoins de l'entreprise ainsi que d'en apprendre un peu plus sur la méthode de travail de ses collaborateurs. En comprenant les besoins spécifiques de Thalès mais également en utilisant la virtualisation des conteneurs avec des outils tel que Podman, il m'est alors possible de répondre aux exigences de sécurité, de performance et de flexibilité, tout en optimisant l'utilisation des ressources et en simplifiant la gestion de l'infrastructure informatique. De plus, cela me permet également, en tant que stagiaire, de découvrir de nouvelles notions ainsi que d’apprendre à les maitriser.

2.3 Préparation au déploiement

Maintenant que j'ai achevé cette étape d'analyser des besoins, je peux passer à la préparation de déploiement de la solution et la phase de test. Lorsque d'un administrateur souhaite ajouter une nouvelle solution ou faire quelques modifications sur une ou plusieurs machines du réseau, il ne peut pas réaliser ces opérations sur les machines en production. Lors de la découverte d'une nouvelle solution, de nombreux problèmes peuvent survenir, jusqu'à potentiellement détruire la machine. Cela serait catastrophique de perdre une machine en production sur laquelle plusieurs dizaines de collaborateurs travaillent. L'administrateur passe alors par une phase de découverte de la solution ainsi qu'une phase de tests où il va expérimenter l'outil sans risque de détruire une machine en production.

2.3.1 Apprentissage de la virtualisation de conteneurs

Comme j'ai pu l'expliquer dans la partie précédente, au sein de Thalès, beaucoup de processus fonctionnent sur des environnements virtualisés et principalement sur des conteneurs Podman. N'ayant jamais fait de virtualisation de conteneurs, il m'a fallu apprendre à les maîtriser : savoir les créer, savoir les modifier, savoir les faire communiquer, savoir les débuguer, etc …

Mon apprentissage commença alors par la découverte de Docker, le leader des outils de management de conteneurs. Il aurait été plus fastidieux de commencer cette découverte des conteneurs par Podman, car, même si les deux solutions sont énormément semblables, la solution de RedHat est moins utilisée et légèrement plus complexe sur certains aspects.

Les cours et documentations sur le sujet sont nombreuses, facilitant ainsi l'apprentissage des conteneurs. Afin d'apprendre à maîtriser ce nouvel outil de virtualisation, je me suis fixé un nouveau mini-projet : la création d'un serveur Apache et d'une base de données communicante. Avec ce mini-projet personnel, je m'offre l'occasion d'apprendre une grande partie des conteneurs, qui me sera amplement suffisante pour la réalisation de ma mission.

Lors de la création d'une machine virtuelle, nous téléchargeons une image du système d'exploitation que nous souhaitons installer. Selon sa complexité, cette image peut être plus ou moins lourde et par conséquent plus ou moins rapide à télécharger. Il faut par la suite installer l'image sur notre VM ce qui prend également un certain temps. Un conteneur fonctionne aussi avec des images. Ces images se trouvent sur le dépôt « docker.io ». On peut y retrouver énormément d'images, créer par des entreprises ou bien des particuliers. Les images des conteneurs étant plus légères, elles sont nettement plus rapides à télécharger qu'une image classique pour VM. Cependant, le réel point fort d'un conteneur c'est qu'on n'a pas besoin d'installer cette image lors du lancement de notre conteneur, l'image n'est pas une image d'un système d'exploitation comme pour les VM mais c'est une copie d'un conteneur. Un fois notre image lancée, notre conteneur est créé.

Les images des conteneurs ne s'arrêtent pas non plus à une simple virtualisation d'une machine par défaut. Lors de la création d'une VM, nous nous retrouvons avec une machine par défaut, sans aucune modification ; avec simplement la configuration de base. Les images de conteneurs peuvent contenir des configurations entières qui permettront le lancement d'un service à leur création. Par exemple, il est possible d'utiliser une image contenant une configuration du service Apache, ce qui permettra de créer un serveur Apache au simple lancé de ce conteneur.

Avec ces diverses images, il est alors possible de créer des applications communicantes très rapidement et avec très peu de modifications, simplement en exécutant quelques conteneurs. Le simple lancement de deux conteneurs contenant un service Apache et une base de données aurait pu permettre de réaliser mon mini-projet, cependant, en procédant ainsi, je n'aurais développé que peu de connaissance et compétences, nécessaires pour ma mission.

Pour approfondir mon apprentissage, j'ai préféré construire mes propres conteneurs qui serviront à mon application communicante. La méthode n'est pas bien différente que précédemment, mais nous pouvons personnaliser nos images. J'ai alors créé mon premier Dockerfile qui est un fichier de configuration d'une image de conteneur. Dans ce fichier, nous précisons de quelle image initiale nous souhaitons que notre conteneur se base puis nous pouvons le personnaliser en ajoutant des paquets ou des commandes à exécuter, ce qui permettra d'avoir un conteneur plus ou moins unique.

Dans mon premier Dockerfile, j'ai donc, depuis une image Debian, ajouter les paquets de configuration du service Apache, ainsi que quelques autres paquets utilitaires et précisé quelques commandes permettant la création d'un utilisateur et son répertoire privé, me permettant ainsi d'avoir un conteneur sur lequel un serveur Apache fonctionne.

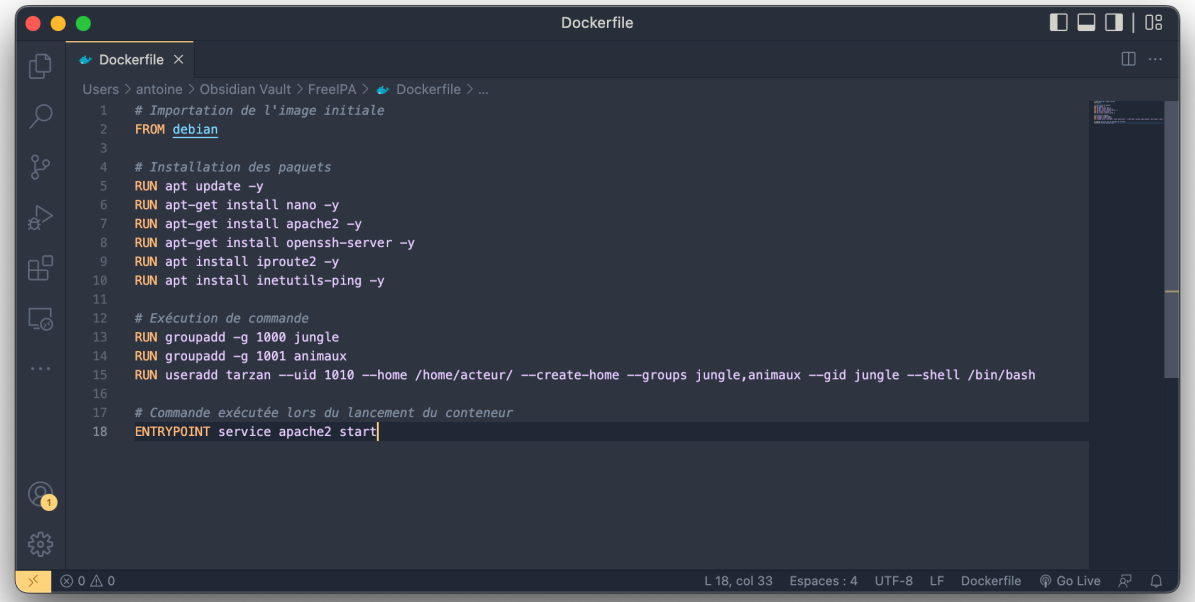

Image 4 : Exemple de contenu d'un fichier Dockerfile

Pour la conteneurisation de la base de données, j'ai rencontré plus de difficultés. Certains services demandes plus de configurations que d'autres, les empêchant de fonctionner correctement sans celles-ci. C'est le cas de la base de données MariaDB que j'ai essayé de créer. La communication avec le serveur Apache que j'avais préalablement créé restait impossible. C'est en examinant le Dockerfile de l'image officielle du conteneur MariaDB que j'ai compris qu'il fallait une certaine configuration pour que cela fonctionne. Il est alors souvent plus sûr et surtout plus efficace d'utiliser une image officielle plutôt que de créer sa propre image.

La création d'un conteneur personnalisé est plus utile lorsque nous avons besoin d'une machine bien spécifique ou pour une machine en production où des mises à jour de paquets sont à faire régulièrement. Dans d'autres cas, il est préférable, si elle existe, d'utiliser une image officielle du service que nous souhaitons mettre en place, nous permettant d'avoir une documentation correcte à disposition et également nous éviter des problèmes de configurations complexes à résoudre.

Il existe un deuxième type de fichier, les docker-compose, qui permettent de configurer des ensembles d'images. Imaginons que je souhaite que mon application communicante soit sur un réseau privé que j'ai créé avec Docker, il faudrait que j'exécute une première commande pour démarrer le premier conteneur avec un ensemble d'option pour préciser au minimum le réseau et le mappage des ports puis faire de même pour le deuxième conteneur. Ces commandes sont longues à taper et sont souvent illisible sur une ligne de commande. Un docker-compose permet de gérer ce genre de cas. Je peux créer un fichier docker-compose dans lequel je précise le mappage de ports, le réseau et toutes les options et

variables d'environnement nécessaires à chacun des conteneurs et en exécutant simplement ce fichier, tous mes conteneurs configurer à l'intérieur de celui-ci démarreront.

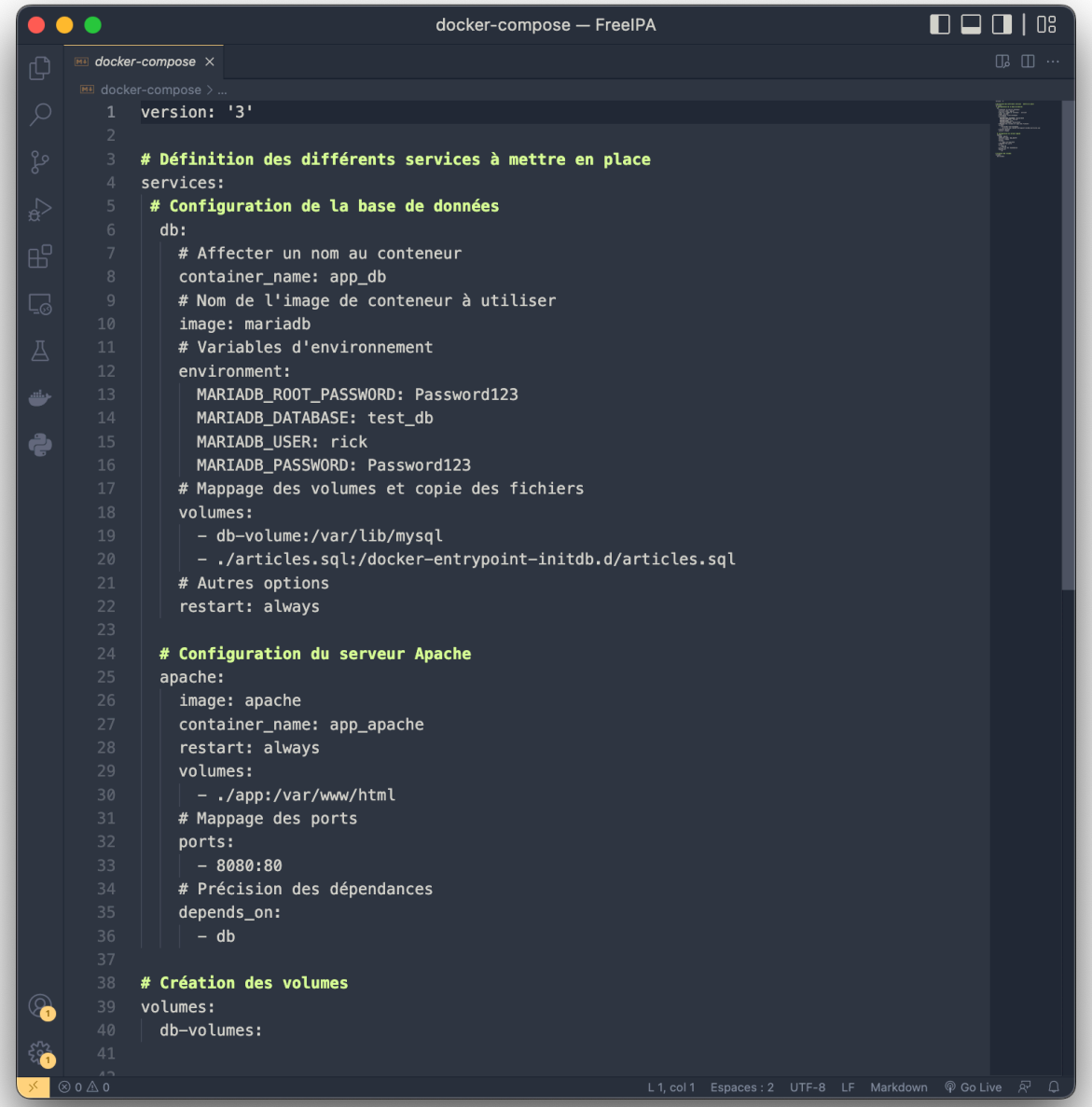

Image 5 : Exemple du contenu d'un fichier docker-compose

2.3.2 Expérimentation de la solution FreeIPA

Après avoir fait connaissance avec les conteneurs, je devais apprendre à connaitre la solution FreeIPA, son fonctionnement et ses fonctionnalités. Par chance, un très grand nombre de documentations sur le sujet ont été publié, ce qui facilite la prise en main de FreeIPA.

Pour pouvoir commencer à prendre en main l'outil et ainsi essayer de découvrir la plupart de ses fonctionnalités, je me suis mis en situation plus ou moins similaire aux conditions du déploiement final. Sur un ordinateur de test hôte, j'ai souhaité configurer une machine virtuelle sur laquelle tournerait le

serveur et une deuxième machine virtuelle pour simuler un client. Cependant, cela fut impossible de configurer deux machines virtuelles sur cet ordinateur dû à ses faibles performances.

J'ai alors pris la décision de réaliser l'expérience en simulant des clients sur de conteneurs Podman. J'ai, pour cette occasion, développer une image de conteneur permettant de faire fonctionner un client FreeIPA sur celui-ci. Cela ne fut pas de tout repos. Étant donné qu'un conteneur partage certains éléments d'une machine avec la machine hôte, à l'opposé d'une machine virtuelle qui est totalement isolée, cela fut un peu difficile de trouver une solution. Après plusieurs heures de test, j'ai pu enfin trouver une solution permettant de déployer un client FreeIPA sur un conteneur.

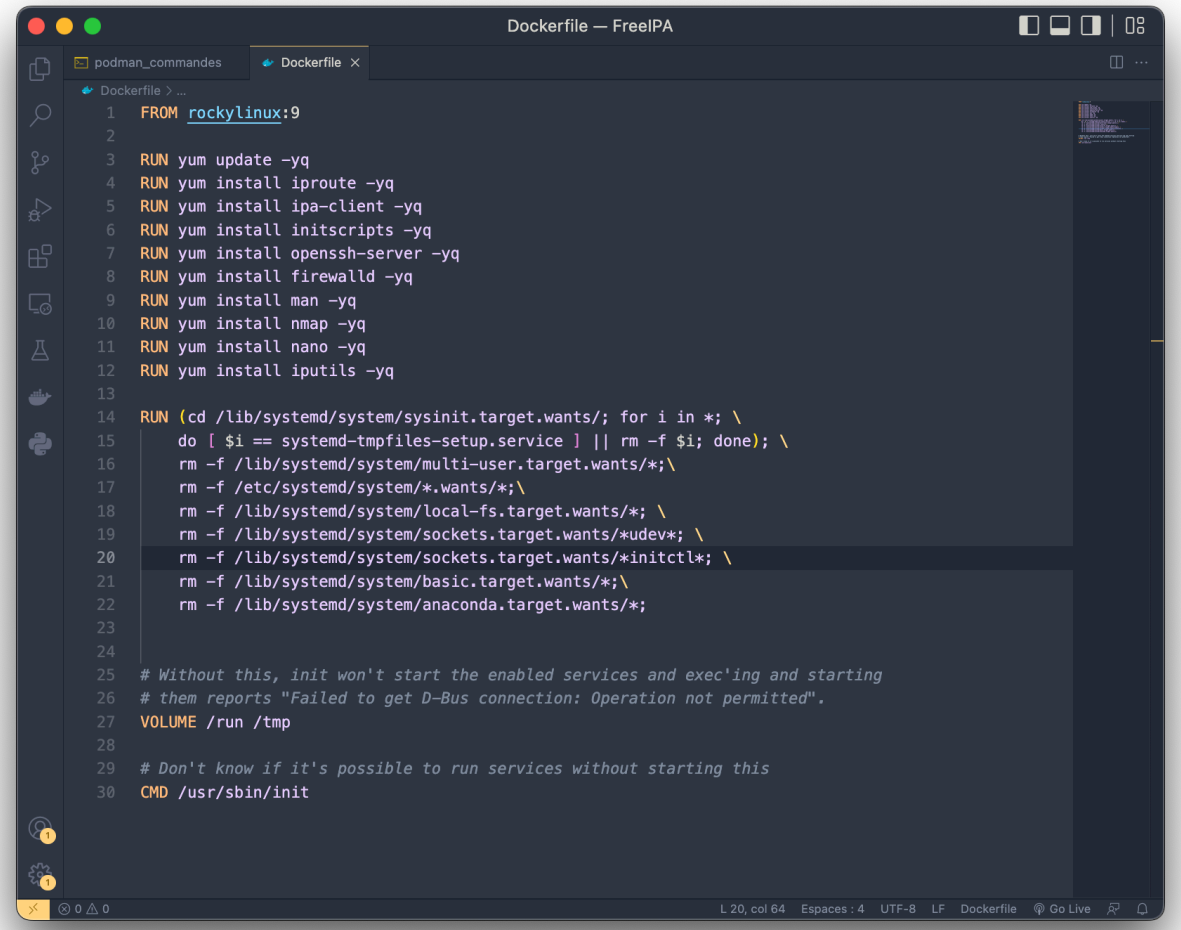

Image 6 : Fichier Dockerfile du conteneur client

La réussite de la création de ce conteneur m'a permis de déployer une machine virtuelle contenant le serveur FreeIPA mais également deux clients situés sur des conteneurs. En effet, la création de ce fichier Dockerfile m'a offert la possibilité de déployer un grand nombre de client au simple lancer d'une commande. Je me suis contenté de deux clients afin d'avoir un environnement de test suffisant pour découvrir la solution.

Par la suite, à l'aide de documentations et de tutoriels en ligne, je me suis entrainé à déployer la solution FreeIPA dans cet environnement. Le déploiement en lui-même n'était pas très complexe, quelques arguments à connaitre sont nécessaires et suffisants. Le plus laborieux est de configurer la solution sur les clients et le serveur pour qu'elle puisse répondre à nos attentes.

J’ai alors réalisé différentes phases de test, permettant de tester et essayer de mettre en place les différentes fonctionnalités de FreeIPA. Les premiers tests ont été la mise en place de règles HBAC, Host Base Access Control, contrôle d’accès basé sur l’hôte en français. Cela permet d’autoriser ou non l’accès à un utiliser ou un groupe d’utilisateur à un client. Il est également possible de configurer un peu plus ces règles pour préciser avec quel protocole l’utilisateur à l’autorisation de se connecter au client.

J’ai par la suite réalisé plusieurs autres expérimentations sur les fonctionnalités de la solution telles que la configuration de règles Sudo (les permissions pour les droits administrateurs), la création de groupes et d’utilisateurs, l’auto-adhésion d’un utilisateur à un groupe ou encore la modification de la politique de mot de passe. Ce sont des fonctionnalités simples à prendre en main et à configurer et donc, par conséquent, je me suis attardé sur des fonctionnalités plus complexes telles que la mise en place d’une double authentification, l’utilisation de clé de sécurité physique permettant l’authentification ou encore l’utilisation des clés publiques SSH.

Ces dernières fonctionnalités ne fonctionnent pas correctement sur tous les systèmes et mes tests m’ont permis de le découvrir. La double authentification est une amélioration de la sécurité intéressante mais qui ne fonctionnait pas sur les clients Ubuntu 22.04. Néanmoins, je ne pense pas que cette fonctionnalité ne soit pas disponible, je pense qu’il existe quelques manipulations à réaliser permettant de les activer que je n’ai pas encore découvert.

Ces expérimentations m'ont permis d'acquérir des connaissances approfondies sur la solution FreeIPA et de tester ses fonctionnalités dans un environnement contrôlé. J'ai également identifié certaines limitations et des problèmes de compatibilité avec certains systèmes, ce qui m'a encouragé à poursuivre mes recherches pour découvrir d'autres manipulations permettant d'activer ces fonctionnalités, telle que l’activation des images de profil utilisateur. En somme, en combinant mes connaissances acquises en virtualisation de conteneurs avec l'expérimentation de la solution FreeIPA, j'ai développé les compétences nécessaires pour mener à bien ma mission de déploiement et d'intégration de cette solution au sein de l'infrastructure de Thalès. Ces apprentissages ont renforcé ma confiance dans ma capacité à gérer les défis liés à la mise en place de solutions technologiques avancées et à répondre aux besoins spécifiques de l'entreprise.

3 Déploiement de la solution FreeIPA

Pour remettre dans le contexte, dans le cadre de la migration du réseau actuel vers un nouveau réseau, je dois mettre en place la solution FreeIPA. Cela permettra d'actualiser les services actuels du réseau mais également de regrouper toutes les informations sur les services, les utilisateurs et sur les hôtes au même endroit.

Maintenant que j'ai pu prendre connaissance des besoins et appris à maîtriser la solution, je peux alors m'attaquer au réel déploiement du serveur et des clients FreeIPA. Pour des raisons de confidentialité, je ne vais pas dévoiler d'information sur le réel déploiement. Je vais expliquer comment celui-ci s'est déroulait en prenant pour exemple une situation similaire. Cela permettra de respecter la confidentialité des noms des utilisateurs, de celui des machines et également des adresses IP.

J'ai également écrit une documentation qui sera plus complète sur comment déployer et configurer quelques fonctionnalités de la solution FreeIPA avec toutes les images de la configuration. Il ne me sera pas possible d'expliquer l'entièreté des étapes suivies dans ce rapport donc je vous renvoie à la documentation du déploiement pour plus de détails (voir annexe 5).

3.1 Installation du serveur FreeIPA

L'installation d'un serveur FreeIPA est relativement simple. Dans notre situation, il est préférable d'installer ce serveur sur un conteneur. Cela permet d'isoler les processus des autres serveurs virtuels fonctionnant sur le même serveur physique et ainsi éviter les conflits.

Comme dit précédemment, la plupart des machines fonctionnent avec l'OS de RedHat, il est alors logique d'utiliser un conteneur Podman, la solution de virtualisation de conteneur soutenue par RedHat, pour déployer notre serveur.

Pour déployer ce serveur, il me suffit de lancer une seule commande comportant toutes les options nécessaires. L'installation du serveur se fera toute seule et l'affichage des fichiers journaux me permettent de suivre l'installation et ainsi vérifier son bon déroulement.

La commande que j'utilise pour l'installation du serveur FreeIPA est relativement longue (voir image ci-après). Pour faciliter le management du serveur en cas d'extinction de celui-ci, il est préférable d'utiliser un fichier docker-compose, permettant d'éviter de réécrire la longue commande utilisée précédemment.

Pour répondre à ce besoin de simplification du management, j'ai écrit un fichier podman-compose, équivalant au fichier docker compose mais fonctionnant avec Podman, qui sera utilisé pour lancer le serveur en utilisant seulement la commande : podman-compose -f <nom-du-fichier> up. Il est bien évidemment plus simple d'utiliser cette commande plutôt que la commande podman run précédente.

```
podman run -dit \
    --name freeipa_app \
    --hostname freeipa.domain.intern \
    --restart always \
    -e IPA_SERVEUR_HOSTNAME=freeipa.domain.intern \
    -e TZ="Europe/Paris" \
    -e SERVER_ADDR=10.88.0.3 \
    --tty=true \
    --cap-add NET_ADMIN \
    -v /etc/localtime:/etc/localtime:ro \
    -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
    -v datafreeipa:/data:Z \
    --sysctl net.ipv6.conf.all.disable_ipv6=0 \
    --security-opt seccomp:unconfined \
    -p 80:80 -p 443:443 -p 389:389 -p 636:636 -p 88:88 \
    -p 464:464 -p 88:88/udp -p 443:443/udp -p 123:123/udp \
    -p 7389:7389 -p 9443:9443 -p 9444:9444 -p 9445:9445 \
    --ip=10.88.0.3 \
    docker.io/freeipa/freeipa-server:rocky-9 \
    -U --domain=domain.intern \
    --realm=domain.intern \
    --http-pin=Password123 \
    --dirsrv-pin=Password123 \
    --ds-password=Password123 \
    --admin-password=Password123 \
    --no-host-dns
```

Image 7 : Commande de déploiement du serveur FreeIPA

```
version: '3.5'

services:
  freeipa_app:
    hostname: ipa
    domainname: ${IPA_SERVER_HOSTNAME}
    container_name: freeipa_server
    restart: unless-stopped
    tty: true
    image: docker.io/freeipa/freeipa-server:rocky-9
    volumes:
      - "/etc/localtime:/etc/localtime:ro"
      - "/sys/fs/cgroup:/sys/fs/cgroup:ro"
      - "datafreeipa:/data:Z"
    environment:
      IPA_SERVER_HOSTNAME: ${IPA_SERVER_HOSTNAME}
      IPA_SERVER_IP: ${IPA_SERVER_ADDRESS}
      TZ: "Europe/Paris"
    command:
      - -U
      - --domain=${IPA_DOMAINNAME}
      - --realm=${IPA_REALM}
      - --admin-password=${IPA_PASSWORD}
      - --http-pin=${IPA_PASSWORD}
      - --dirsrv-pin=${IPA_PASSWORD}
      - --ds-password=${IPA_PASSWORD}
      - --no-host-dns
    cap_add:
      - NET_ADMIN
    sysctls:
      - net.ipv6.conf.all.disable_ipv6=0
      - net.ipv6.conf.lo.disable_ipv6=0
    security_opt:
      - "seccomp:unconfined"
    networks:
      ipa-network:
        ipv4_address: ${IPA_SERVER_ADDRESS}
        aliases:
          - freeipa

networks:
  ipa-network:
    driver: macvlan
    name: ipa-network
    external:
    ipam:
      config:
        - subnet: ${IPA_NETWORK_SUBNET}
```

Image 8 : Fichier podman-compose utilisé pour le déploiement

En suivant l'avancement de l'installation du serveur en utilisant les fichiers logs, je peux voir si l'installation du serveur s'est déroulée correctement. Lorsque l'installation est terminée, je peux me connecter à l'interface web, qui me permettra par la suite de faire la plupart des configurations du serveur et de ses clients.

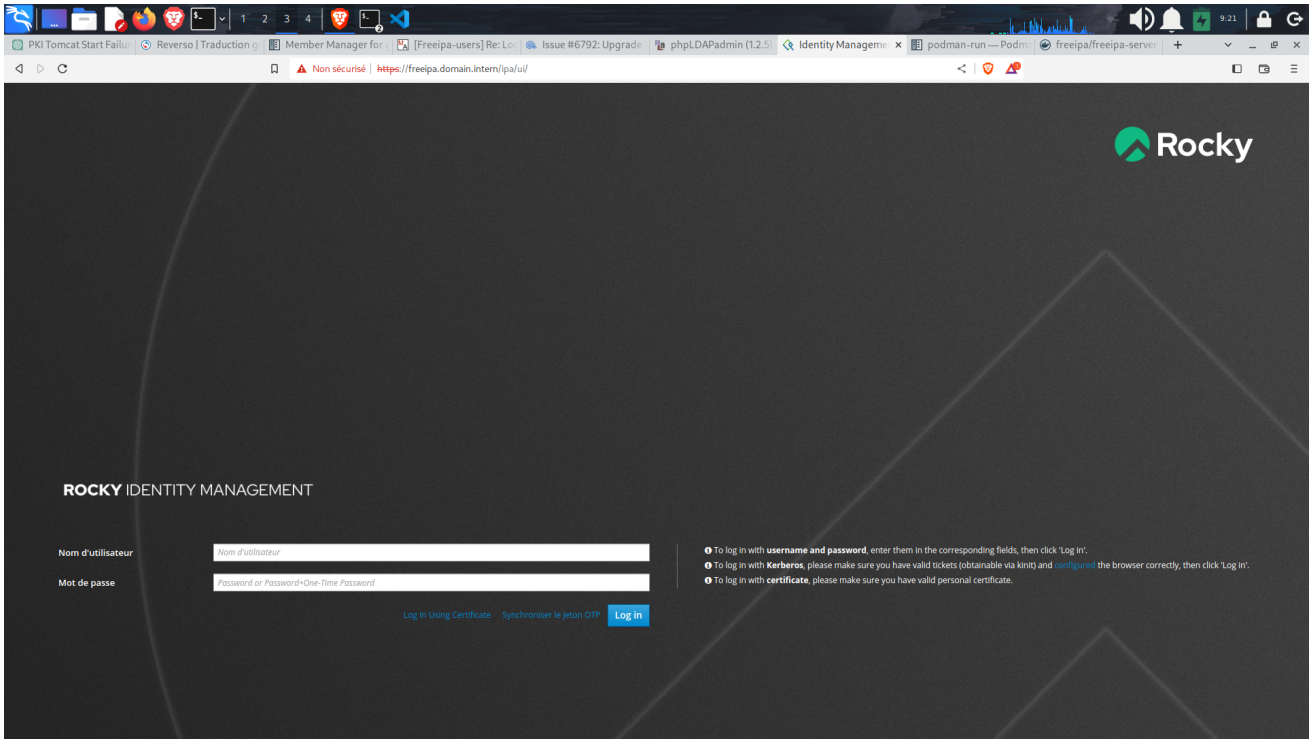

Image 9 : Interface web du serveur

3.2 Installation du client FreeIPA

À l'instar du serveur, l'installation du client est également relativement simple. Pour pouvoir procéder à celle-ci, il est nécessaire de commencer par l'installation du paquet de configuration : ipa-client ou freeipa-client, selon le système d'exploitation utilisé.

Pour que le client puisse se configurer, il est également nécessaire que la machine client puisse communiquer avec le serveur mais aussi faire sa résolution de nom, c'est-à-dire trouver la correspondance entre le nom du serveur et son adresse IP. Pour cela, soit le serveur est inscrit dans le DNS, comme c'est le cas chez Thalès et alors je n'ai rien à faire, soit ce n'est pas le cas et je dois l'ajouter moi-même. Je peux aussi l'ajouter dans le fichier **/etc/hosts** si je n'ai pas de serveur DNS, ce qui permet de faire la résolution du nom en local.

Une fois que mes préparatifs sont réalisés, je peux lancer la commande pour la configuration du client. Je peux, selon mes envies soit réaliser la configuration en suivant l'affichage de celle-ci et entrer les données nécessaires lorsqu'elles me sont demandées, soit je peux donner toutes les données nécessaires en option de la commande de lancement de la configuration du client.

Pour l'installation du client j'ai préféré utiliser la configuration avec toutes les données entrées en option de la commande de lancement, ce qui permet également d'ajouter quelques options de configuration dès l'installation du client et éviter de le faire plus tard, tel que l'option –mkhomedir, l'option qui permet la création des espaces personnels des utilisateurs lors de leur première authentification sur le client.

Image 10 : Commande de configuration du client

Évidemment, lors du réel déploiement du serveur, je n'ai pas utilisé les options -p et -w permettant de préciser le login et le mot de passe pour des questions de sécurité. Il a également fallu que je modifie et rajoute quelques options pour que le client puisse être configuré dans l'architecture réseau de Thalès.

Une fois la commande exécutée et que la configuration est réussite, je peux me rendre sur l'interface Web afin de vérifier que le nouveau client s'affiche correctement dans l'interface. C'est bien le cas, je peux donc continuer mon déploiement et ainsi configurer le reste des clients.

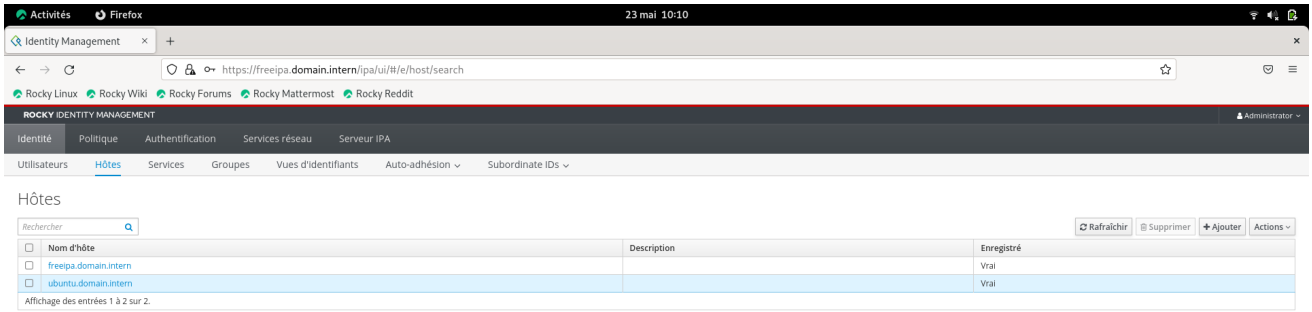

Image 11 : Liste des clients et serveurs sur l'interface Web

L'actuel serveur LDAP étant sur le réseau « rouge », le réseau isolé, les machines du réseau « blanc » ne peuvent pas communiquer avec celui-ci. Il existe une méthode pour migrer tous les utilisateurs et les groupes depuis un serveur LDAP vers le serveur FreeIPA qui nécessite que ces deux serveurs puissent communiquer. J'expliquerai dans la partie suivante comment faire cette migration mais dans ma situation cela n'a pas était possible. On m'a alors demandé d'ajouter les groupes et les utilisateurs à partir des fichiers **/etc/passwd** et **/etc/groups**, les fichiers contenant les informations sur les utilisateurs et les groupes.

Pour ajouter ces groupes et ces utilisateurs, j'ai alors conçu un programme Python permettant de lire ces fichiers de descriptions et d'ajouter les utilisateurs et les groupes sur le serveur FreeIPA de manière quasiment automatique. Il peut y avoir quelque exception qui seront traitées manuellement par la suite. Ce script Python génère également de nouveaux mots de passe temporaire pour les utilisateurs, qui

sont inscrits dans un fichier pour que je puisse transmettre le bon mot de passe au bon utilisateur. Un second fichier est créé avec tous les utilisateurs qui n'ont pas été ajoutés, souvent les utilisateurs systèmes mais quelques exceptions peuvent être possibles.

```python
#! /bin/bash
import os
import sys
from string import ascii_letters, digits
import random

# File path
path_group = "/etc/group"
path_users = "/etc/passwd"

# Password variable
characters = ascii_letters + digits
length_pwd = 12
default_shell = '/bin/bash'
gid_min = 1000
gid_max = 50000

def main():

    try:
        # Assert validity of files
        os.path.isfile(path_users)
        assert os.stat(path_users).st_size > 0
        os.path.isfile(path_group)
        assert os.stat(path_group).st_size > 0

    except Exception as e:
        # Print error
        print("Unexpected error occurred: {}".format(e))
        sys.exit(1)

    else :
        # Open /etc/passwd file
        with open(path_users, 'r') as file_users :
            # Read file
            data = file_users.readlines()
            for line in data:
                # Line traitement
                line = line.split(":")
                if len(line) == 7:
                    user = {"login": line[0], "uid": line[2], "gid": line[3], "gecos": line[4], "homedir": line[5], "shell": line[6].strip()}
                    name = user['gecos'].split(" ")
                    # If it is a potentially real user
                    if user["shell"] != '/sbin/nologin':

                        if len(name) == 2  and int(user["uid"]) >= 1000:
                            first_name, last_name = name
                            password = ''.join(random.choice(characters) for i in range(length_pwd))
                            # Add user
                            os.system(f"echo '{password}' | ipa user-add {user['login']} --uid={int(user['uid'])} --gidnumber={int(user['gid'])} --
cn='{user['gecos']}' --first='{first_name}' --last='{last_name}' --shell={default_shell} --homedir='{user['homedir']}' --password")
                            # Write user added logs
                            with open("./users_added", 'a') as file:
                                file.write(','.join([user['login'],user['gecos'],user['uid'],password, '\n']))
                        # If gecos is not correct
                        else :
                            # Write user not added logs
                            with open("./users_not_added", 'a') as file:
                                file.write(','.join([user['login'],user['uid'],user['gid'],user['gecos'],user['homedir'],user['shell'],'\n']))


        # Open /etc/group file
        with open(path_group, 'r') as file_group :
            # Read file
            data = file_group.readlines()
            for line in data :
                # Line traitement
                line = line.split(":")
                # If line is correct
                if (len(line)) == 4 :
                    group = {"group_name": line[0], "gid": line[2], "members": line[3]}
                    # If GID is between the gid_min and the gid_max supply
                    if int(group["gid"]) > gid_min and int(group["gid"]) < gid_max:
                        # Add group
                        os.system(f"ipa group-add {group['group_name']} --gid {int(group['gid'])}")
                        # If members in group
                        if group["members"] != '\n' :
                            list_members = (line[3].rstrip("\n")).split(",")
                            members_list_cmd = ""
                            for member in list_members :
                                # Add member in group
                                members_list_cmd += f" --user={member} "
                            os.system(f"ipa group-add-member {group['group_name']} {members_list_cmd}")


if __name__ == '__main__' :
    main()
```

Image 12 : Script Python

3.3 Configuration de la solution FreeIPA

Maintenant que tous mes clients et mon serveur FreeIPA ont bien été installés et affichés sur l'interface Web, je peux passer à la configuration. Comme dit plus tôt, FreeIPA offre énormément de fonctionnalités. Je n'ai évidemment pas pu tester toutes ces fonctionnalités mais je me suis focalisé sur celle qui était le plus intéressante à mettre en place.

Pour commencer, je vais vous expliquer comment fonctionne la migration via un serveur LDAP que j'ai énoncé précédemment. Si le serveur LDAP et le serveur FreeIPA peuvent communiquer, alors cette fonctionnalité est possible. Pour se faire, il me suffit d'activer sur le serveur FreeIPA le module de migration avec la commande **ipa config-mod --enable-migration=TRUE**. Il m'est alors désormais possible de migrer les informations du serveur LDAP vers le serveur FreeIPA, en précisant les informations que je souhaite transférer.

Image 13 : Commande de migration LDAP vers FreeIPA

Il est important de noter que lorsque j'ai utilisé cette commande, tous les utilisateurs et les groupes ont bien été transférés mais les utilisateurs n'ont pas été affectés à leurs groupes. Pour traiter ce problème, j'ai conçu un autre script Python permettant d'ajouter les utilisateurs dans leurs groupes depuis un fichier provenant du serveur LDAP (voir annexe 6).

Une fois que tous mes groupes et utilisateurs ont bien été configurés sur mon serveur FreeIPA, je peux commencer à mettre en place les premières règles de sécurité. Les règles HBAC, pour rappel, Host Based Access Control, sont donc des règles permettant le contrôle d'accès. Il m'est possible d'autoriser ou non l'accès à un client ou à un serveur uniquement pour certains utilisateurs. Ces règles sont nécessaires dans un environnement où les collaborateurs n'ont pas tous les mêmes droits d'accès selon la confidentialité des projets.

En me rendant sur l'interface Web, je peux alors mettre en place de ces règles. Leur configuration est très simple. Je sélectionne les utilisateurs et les hôtes pour lesquels je veux que ma règle s'applique, et je peux également préciser les protocoles d'accès autorisés. Une fois ma règle créée, je peux désactiver la règle par défaut qui autorise tous les utilisateurs à accéder à toutes les machines, et tester si ma règle fonctionne correctement.

Bien utilisé, un ensemble de ces règles permettent de garantir une gestion des degrés de confidentialité optimum. De plus, il est très simple et rapide d'ajouter des nouveaux collaborateurs sur un projet et dans ce cas-là lui donner les accès nécessaires en l'ajoutant simplement dans un groupe de projet pour lequel une règle HBAC existe déjà.

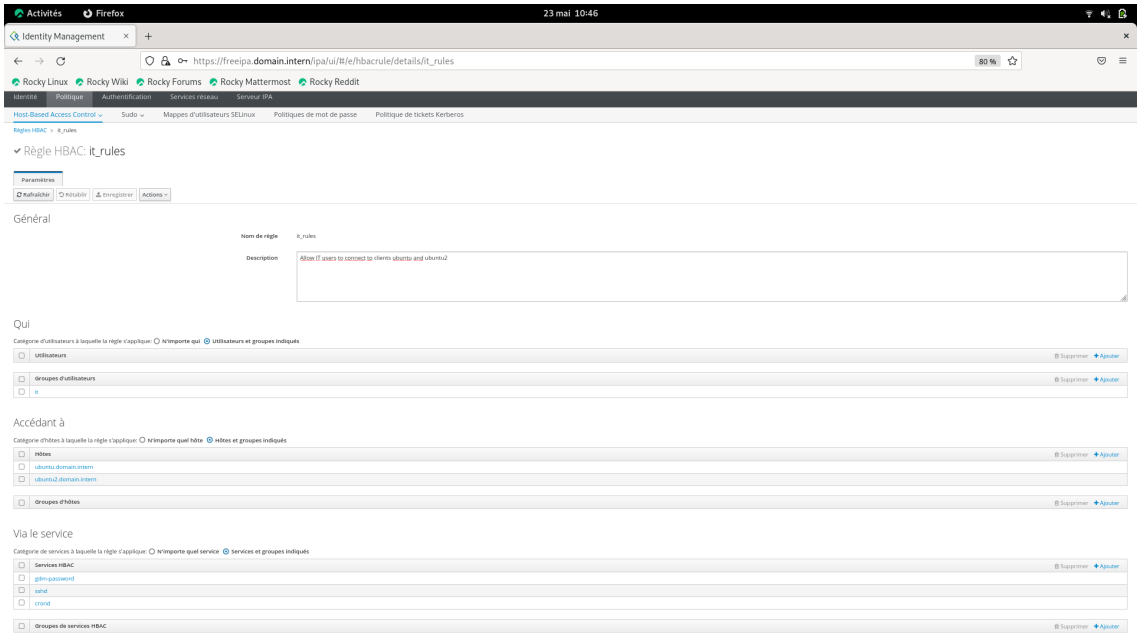

Image 14 : Interface de configuration de règle HBAC

La seconde fonctionnalité importante est la possibilité de créer des règles Sudo. Ce sont les règles permettant d'autoriser des commandes nécessitant des droits administrateurs. Actuellement, aucun utilisateur, même les comptes des administrateurs n'ont la possibilité d'utiliser de telles commandes, les administrateurs sont obligés de se connecter sur le compte de super-administrateur (root) pour pouvoir utiliser ces commandes.

Ces règles permettre alors d'autoriser par exemple le compte d'un administrateur d'utiliser ces commandes sans permettre aux autres utilisateurs de le faire. De plus, ces règles n'autorisent pas l'utilisation de toutes les commandes mais uniquement de celles spécifier, ce qui permet une gestion des droits une fois de plus optimale. Autoriser les comptes utilisateurs, principalement ceux des administrateurs, permet de limiter l'utilisation du compte root mais également d'avoir une trace de quel utilisateur a utilisé quelle commande.

Ces règles sont cependant un peu plus complexes et fastidieuses à mettre en place, par rapport à une règle HBAC, mais reste encore relativement simple. Je vous renvoie à la documentation que j'ai écrite pour la configuration de telle règle (voir annexe 5).

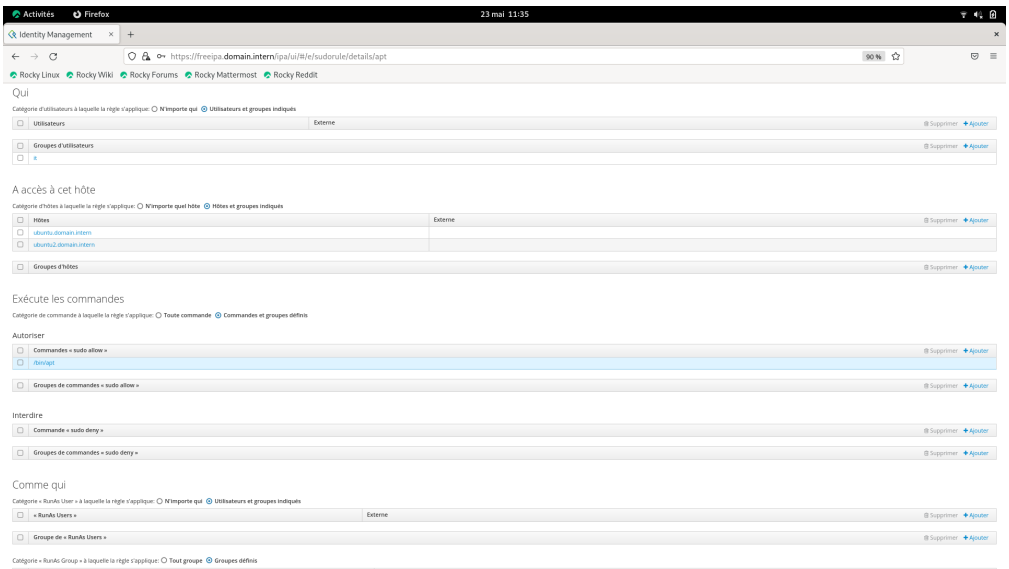

Image 15 : Interface de configuration de règle Sudo

Je me suis attaqué ensuite à la modification de la politique des mots de passe. Comme je l'ai précisé plus tôt, les mots de passe sont aujourd'hui facilement trouvés en quelques minutes s'ils ne sont pas forts. En modifiant la politique par défaut, je peux alors obliger les collaborateurs à utiliser des mots de passe forts. De plus, je peux obliger le changement de mot de passe régulièrement, tous les 3 mois par exemple, offrant alors une meilleure sécurité et une meilleure gestion des mots de passe. Cette politique est très complète et facilement modifiable vie l'interface Web, permettant alors, en quelques clics d'accroître la sécurité.

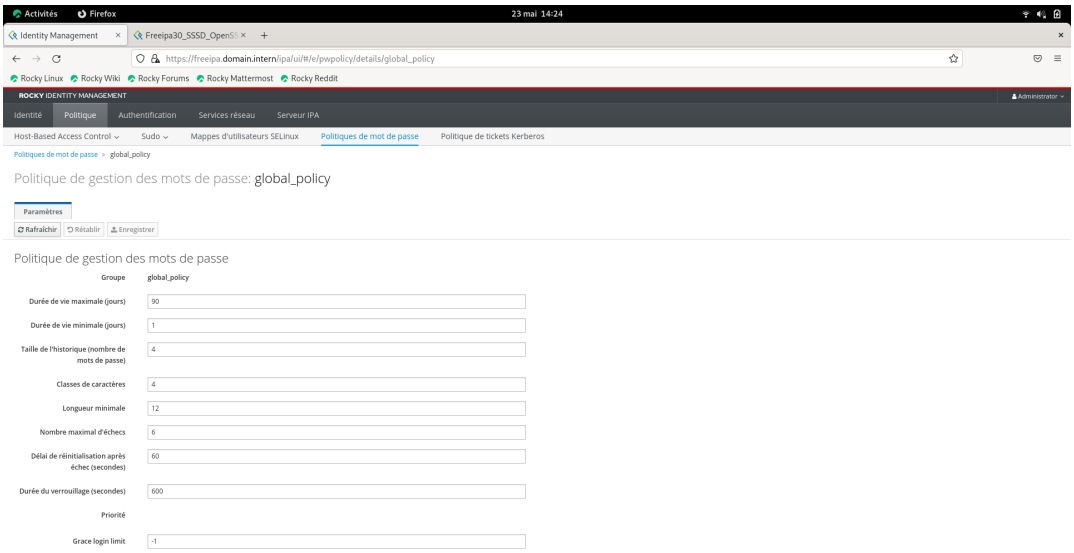

Image 16 : Configuration de la politique des mots de passe

Un autre moyen de s'authentifier est l'utilisation de clés SSH. Le protocole SSH permet de se connecter à distance sur une machine, en utilisant par défaut un mot de passe et un login. Il est possible de générer une paire de clés, publique et privée, permettant de s'authentifier sans mot de passe sur une machine. La solution FreeIPA permet de gérer les clés SSH publiques des utilisateurs et alors permettra la connexion à distance d'un utilisateur sur n'importe quel client avec la même clé privé. Normalement, il faudrait copier la clé publique de l'utilisateur sur tous les clients mais avec FreeIPA il faut simple l'ajouter sur le profil de l'utilisateur.

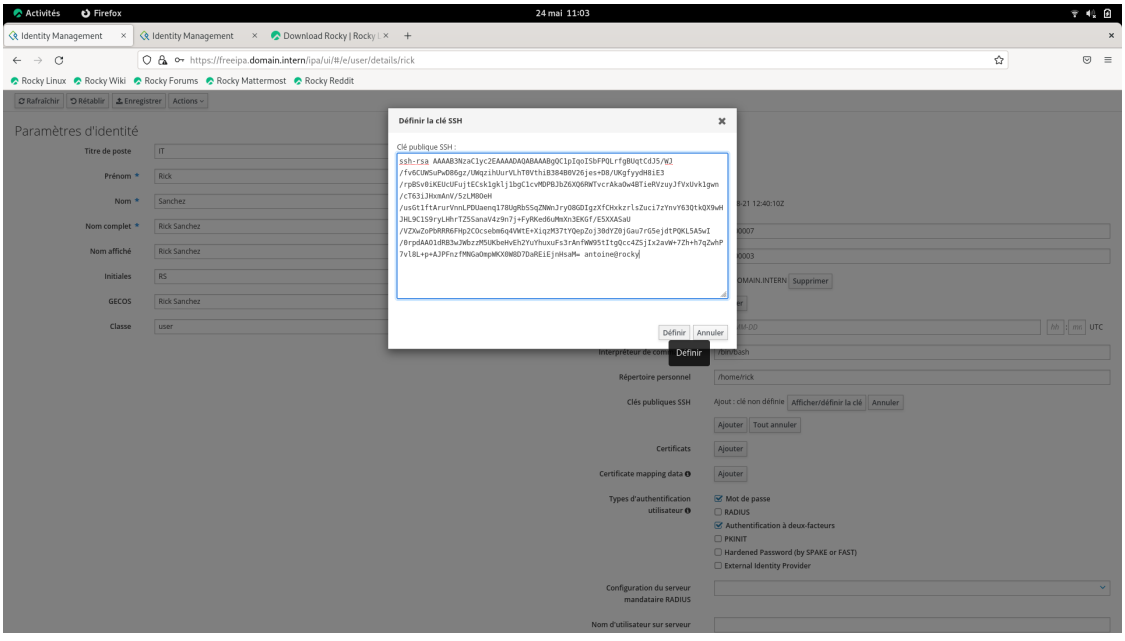

Image 17 : Configuration de l'authentification par clé SSH

Ces quatre fonctionnalités sont importantes, voir essentielles pour le bon fonctionnement mais surtout pour garantir la sécurité du réseau. Elles sont extrêmement simples à mettre en place et augmentent énormément la sécurité du réseau. Il m'était alors inévitable de les utiliser. J'ai également testé d'autre fonctionnalité telle que la double authentification, que je n'ai pas forcément mis en place.

Lors de mes tests sur la double authentification, j'ai pu remarquer que celle-ci ne fonctionnait pas avec tous les systèmes d'exploitation. Il me serait alors compliqué et conflictuel de forcer l'utilisation de la double authentification. Celle-ci reste toujours possible si un utilisateur souhaite l'utiliser. FreeIPA permet de générer un QR code, qui une fois scanner avec une application adéquate telle que Google Authentificator, génère un mot de passe temporaire à saisir après le mot de passe de son compte utilisateur. Pour pouvoir utiliser cette fonctionnalité, je dois également l'autoriser soit sur le compte de l'utilisateur pour permettre seulement à celui-ci de l'utiliser, soit sur un client pour obliger tous les utilisateurs souhaitant se connecter au client à l'utiliser.

Enfin, j'ai également mis en place une extension au serveur FreeIPA pour personnaliser un peu plus le profil des utilisateurs. Sur le serveur LDAP actuel, il est possible d'ajouter et visualiser une photo de profil de l'utilisateur, permettant alors d'avoir un profil plus complet d'un collaborateur. Nativement, cette fonctionnalité n'est pas disponible sur FreeIPA. Il m'est possible d'ajouter une photo dans le profil utilisateur mais elle n'est pas affichée sur l'interface Web.

J'ai alors ajouté au serveur une extension créer par un membre de la communauté permettant d'ajouter sur l'interface Web un onglet pour cette photographie. La documentation de son activation étant bien expliquée, il m'a été très simple de la mettre en place en exécutant le fichier d'installation sur le serveur FreeIPA.

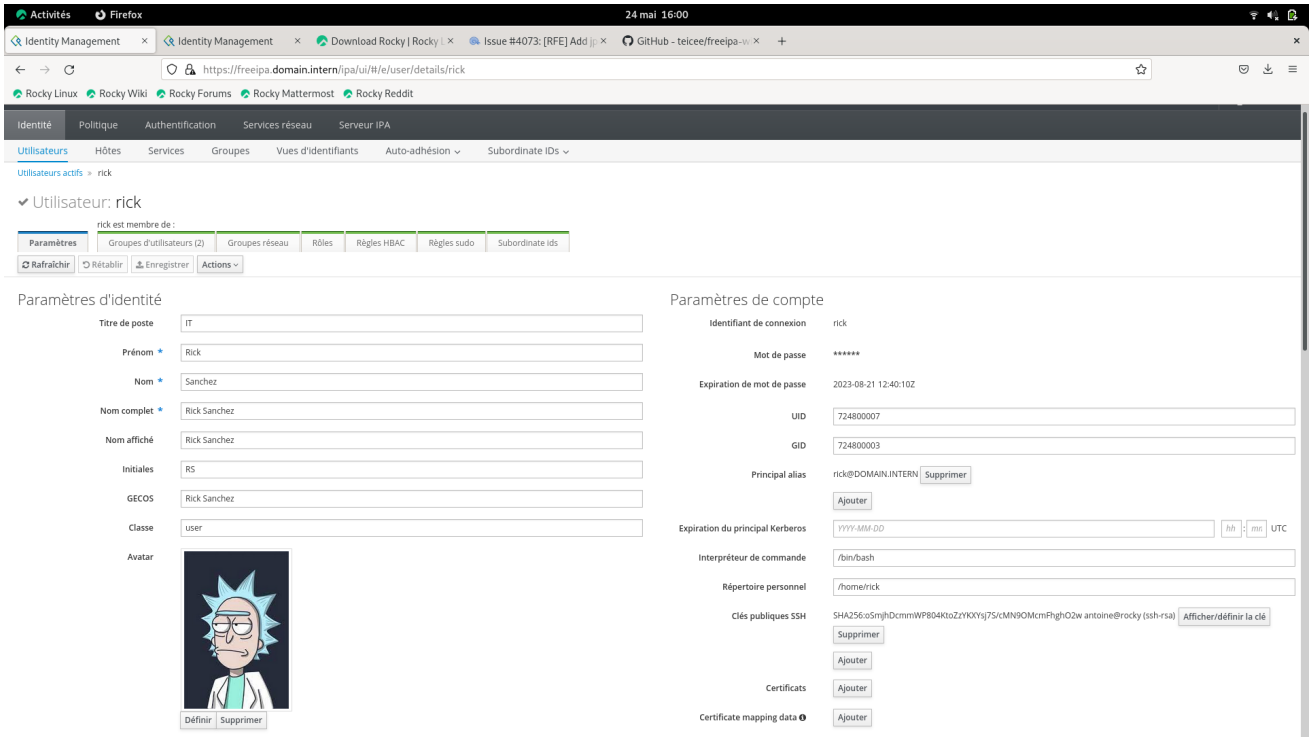

Image 18 : Profil utilisateur avec photo

Il reste énormément de fonctionnalité à découvrir et potentiellement mettre en place mais, par faute de temps, je ne pourrais pas les tester et les mettre en place. J'ai déjà pu mettre ne place les fonctionnalités essentielles pour le bon fonctionnement et la sécurité du réseau, et ainsi répondre aux objectifs demandés initialement, avec quelques petites fonctionnalités en plus. La solution FreeIPA que j'ai mis en place est alors opérationnelle, mais n'est pas encore active sur les machines en production. Cela ne sera pas fait par mes soins, mais réaliser par les administrateurs de l'entreprise.

4 Conclusion

Au cours de ce stage, j'ai eu l'occasion de découvrir un peu plus le monde d'un administrateur système et ainsi développer des compétences qui lui sont propres. J'y ai pu découvrir de nouvelles notions tel que les conteneurs, ou approfondir les notions appréhendées en cours. La mise en place de cette solution client-serveur FreeIPA m'a donc permit d'approfondir mes compétences mais également me mettre à l'épreuve, tester mes compétences dans le domaine et visualiser si elles sont suffisantes pour des missions de ce type. La réussite de ma mission m'a alors convaincu que j'avais les capacités pour une telle mission. Je reste tout de même sceptique sur ma quantité de connaissance, j'ai pu constater lors de discussion de l'écart de savoir entre un professionnel et un étudiant, mais ne désespère pas car ce sont des connaissances et des compétences qui se développent avec les temps et la pratique.

Enfin, ce stage m'a permis également de découvrir un métier du monde des réseaux et télécommunications. Cette première expérience dans ce domaine fut satisfaisante et enrichissante, sur le plan professionnel et personnel. Celui-ci m'a permis de répondre un peu plus à mes questionnements sur mon plan professionnel et personnel et sur la carrière que je souhaiterais mener.

Je suis tout de même légèrement déçu de ne pas avoir pu mettre en place la solution de contrôle des périphériques proposée par la société Ivanti. Celle-ci était très intéressante et aurait été un atout supplémentaire dans la sécurité de l'entreprise, cependant, même après plusieurs relances, les commerciaux de la société Ivanti n'ont pas fait suite à notre demande.

5 Remerciements

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à tous les collaborateurs de l'entreprise Thalès de m'avoir accueilli au sein de leurs locaux à Meyreuil.

Tout d'abord, je souhaite remercier mon maître de stage, Franck Morier, pour m'avoir accueilli et pris sous sa tutelle. Je tiens également à remercier Jean-Pierre Mellano et Guillaume Denis de m'avoir encadré au sein de l'équipe IT. Leur accueil chaleureux, leur collaboration et leur soutien constant ont grandement contribué à mon apprentissage.

Mes remerciements vont également à mes collègues de travail, qui ont partagé leurs connaissances et leur expérience avec moi. Leur convivialité, leur esprit d'équipe et leur patience ont créé un environnement propice à l'apprentissage et à l'épanouissement professionnel. Leurs conseils et leurs encouragements ont été d'une valeur inestimable.

Je souhaite exprimer ma gratitude envers mon établissement d'enseignement, l'IUT Réseaux et Télécommunication de Luminy, pour m'avoir offert l'opportunité de réaliser ce stage. Je suis reconnaissant envers mes enseignants et mes mentors qui m'ont préparé et soutenu tout au long de ma formation, me permettant ainsi d'appliquer mes connaissances dans un contexte réel.

Mes plus sincères remerciements à tous.

Antoine Planas

6 Glossaire

DIS, Digital Identity & Security

BUT, Bachelor Universitaire de Technologie

Backup, Une sauvegarde (backup en anglais) est une copie de données, de fichiers ou d'un système informatique effectuée dans le but de prévenir la perte de données en cas de défaillance, d'erreur humaine, de sinistre ou d'autres événements indésirables.

Audit, Un audit est un processus d'évaluation et de vérification systématique des activités, des procédures ou des systèmes afin d'identifier les faiblesses, les erreurs ou les non-conformités.

Open-source, Un logiciel open-source est un logiciel dont le code source est disponible et accessible au public. Il peut être librement utilisé, modifié et distribué par quiconque selon les termes de la licence open-source correspondante.

LDAP, Lightweight Directory Access Protocol

DNS, Domain Name Server

Certificat, Un certificat est un fichier numérique qui atteste l'authenticité et la légitimité d'une entité, telle qu'un site web ou une personne

SSO, Single Sign-on

OS, Operating System

Système d'exploitation, Un système d'exploitation est un logiciel qui gère les ressources d'un ordinateur et permet aux utilisateurs d'interagir avec le matériel et les logiciels. Il fournit une interface entre les utilisateurs et le matériel informatique, facilitant l'exécution de programmes et la gestion des fichiers et des périphériques. Exemples de systèmes d'exploitation : Windows, MacOs, Linux, etc.

AD, Active Directory

Paquet, Un paquet, dans le contexte informatique, fait référence à un ensemble de fichiers regroupés et compressés en un seul fichier, facilitant leur distribution et leur installation. Les paquets sont couramment utilisés dans les systèmes d'exploitation pour installer des logiciels ou des mises à jour. Ils peuvent contenir les fichiers nécessaires à l'exécution d'un programme ainsi que des informations sur sa version, ses dépendances et sa configuration.

Client, Dans le contexte informatique, un client fait référence à une application, un programme ou un dispositif qui accède aux services ou aux ressources fournis par un serveur. Le client envoie des requêtes au serveur et reçoit les réponses correspondantes.

VM, Virtual Machine

Processus, Un processus est une instance d'un programme en cours d'exécution dans un système d'exploitation. Il peut être considéré comme une tâche ou un ensemble de tâches qui sont exécutées en parallèle ou séquentiellement.

Sandbox, Une sandbox, ou bac à sable, est un environnement d'exécution isolé et sécurisé dans lequel les applications ou les processus peuvent s'exécuter sans affecter le reste du système. Cela permet de limiter les risques de sécurité en restreignant les privilèges et l'accès aux ressources sensibles.

Débuguer, Débuguer fait référence au processus de recherche et de correction des erreurs, des bugs ou des problèmes dans un programme informatique. Cela implique généralement l'analyse du code source, l'identification des erreurs et la mise en place de correctifs appropriés pour assurer un fonctionnement correct et fiable du programme.

Apache, Apache, ou Apache HTTP Server, est un logiciel serveur web populaire et open-source.

MariaDB, MariaDB est un système de gestion de base de données open-source qui est une bifurcation (fork) de MySQL. Il offre une solution de base de données relationnelle puissante et évolutive, compatible avec les applications et les outils existants.

Mappage de ports, Le mappage de ports (port mapping en anglais) est une technique utilisée dans les réseaux informatiques pour rediriger le trafic réseau d'un port spécifique vers un autre port. Cela permet de faire correspondre les ports utilisés par les applications sur un réseau interne aux ports utilisés par le réseau externe ou par d'autres applications.

Variables d'environnement, Les variables d'environnement sont des valeurs spécifiques définies dans le système d'exploitation qui peuvent être utilisées par les programmes informatiques en cours d'exécution. Elles fournissent des informations sur l'environnement de travail du programme, telles que les chemins d'accès aux fichiers, les paramètres de configuration, les informations système, les préférences utilisateur, etc.

HBAC, Host Based Access Control

Sudo, Sudo est une commande utilisée dans les systèmes d'exploitation de type Unix, tels que Linux, pour exécuter des commandes avec les privilèges de superutilisateur (root).

SSH, SSH (Secure Shell) est un protocole de communication sécurisé utilisé pour l'accès distant aux systèmes informatiques et l'exécution de commandes à distance. Il fournit un canal crypté pour la transmission des données et permet une authentification sécurisée des utilisateurs.

7 Bibliographie

Conteneurs

- Oracle. Container Registry [en ligne]. Consulté le 17 avril 2023. What is Docker? https://www.oracle.com/fr/cloud/cloud-native/container-registry/what-is-docker/
- NetApp. DevOps Solutions [en ligne]. Consulté le 25 avril 2023. What are Containers? https://www.netapp.com/fr/devops-solutions/what-are-containers/
- Journal du Net. Dictionnaire du Webmastering [en ligne]. Consulté le 2 mai 2023. Conteneur en informatique - Définition précise et fonctionnement.https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445240-conteneur-en-informatique-definition-precise-et-fonctionnement/
- Atlassian. Microservices [en ligne]. Consulté le 10 mai 2023. Containers vs VMs. https://www.atlassian.com/fr/microservices/cloud-computing/containers-vs-vms
- Amazon Web Services. What Is Virtualization? [en ligne]. Consulté le 18 mai 2023. Virtualization - Amazon Web Services (AWS). https://aws.amazon.com/fr/what-is/virtualization/
- Red Hat. Topics [en ligne]. Consulté le 26 mai 2023. What Is Podman? https://www.redhat.com/fr/topics/containers/what-is-podman
- Proofpoint. Threat Reference [en ligne]. Consulté le 13 juin 2023. Sandbox. https://www.proofpoint.com/fr/threat-reference/sandbox

FreeIPA

- FreeIPA. Main Page [en ligne]. Consulté le 17 avril 2023. https://www.freeipa.org/page/Main_Page
- Worteks. Blog [en ligne]. Consulté le 25 avril 2023. FreeIPA-P1. https://www.worteks.com/blog/FreeIPA-P1/
- FreeIPA. Using Yubikey 4 Nano to authenticate to FreeIPA enrolled host [en ligne]. Consulté le 2 mai 2023. https://www.freeipa.org/page/Using_Yubikey_4_Nano_to_authenticate_to_FreeIPA_enrolled_host

Kerberos

- Wikipédia. Kerberos (protocole) [en ligne]. Consulté le 17 avril 2023. https ://fr.wikipedia.org/wiki/Kerberos_(protocole

LDAP / Active Directory

- Okta. Identity 101 [en ligne]. Consulté le 25 avril 2023. LDAP vs. Active Directory. https://www.okta.com/fr/identity-101/ldap-vs-active-directory
- Red Hat. Topics [en ligne]. Consulté le 10 mai 2023. What is LDAP Authentication? https://www.redhat.com/fr/topics/security/what-is-ldap-authentication
- IONOS. Digital Guide [en ligne]. Consulté le 18 mai 2023. LDAP. https://www.ionos.fr/digitalguide/serveur/know-how/ldap/

Single Sign-On

- Oracle. Sécurité [en ligne]. Consulté le 26 mai 2023. Qu'est-ce qu'un SSO (Single Sign-On)? https://www.oracle.com/fr/security/qu-est-ce-qu-un-sso.html
- OneLogin. Learn [en ligne]. Consulté le 13 juin 2023. How Single Sign-On Works. https://www.onelogin.com/fr-fr/learn/how-single-sign-on-works

- Journal du Net. Dictionnaire du Webmastering [en ligne]. Consulté le 13 juin 2023. SSO (Single Sign-On) - Définition, traduction et acteurs. https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203483-sso-single-sign-on-definition-traduction-et-acteurs/