



Rapport final

Dans le cadre du projet pleiade

Auteurs :

Anthony BILLETTE
Jean-Baptiste PESLERBES
Théo PORTIER
Simon RUFFET
Anthony YAR

Encadrants :

Jean-François BELLANGER
Jean-Pierre AUBIN

Version 1.2 du
28 mars 2019

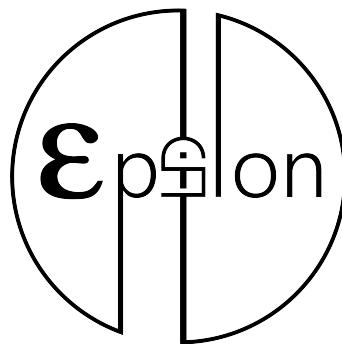


Table des matières

I Le projet Pleiade	
Pilotage Logiciel d'Équipement Informatique Autonome Distant et Embarqué	7
1 Introduction	10
1.1 La genèse	10
1.2 Notre projet	11
2 Gestion de projet	12
2.1 Notre organisation	12
2.2 Nos Réunions	12
2.3 Notre salle	13
2.4 Notre Gantt	13
3 État de l'art	16
3.1 VPN	16
3.1.1 Qu'est-ce qu'un VPN ?	16
3.1.2 Pourquoi utiliser un VPN ?	17
3.1.3 Les différents VPN open-source :	17
3.1.4 Conclusion	18
3.2 Plug-in Firefox	19
3.2.1 Qu'est-ce qu'un plug-in ?	19
3.2.2 Quelle est la structure d'un plug-in ?	19
3.2.3 Conclusion	20
3.3 La conteneurisation	20

3.3.1	Les besoins d'EPSILON :	20
3.3.2	En quoi ça consiste ?	21
3.3.3	Les outils de conteneurisation :	22
3.3.4	Conclusion	23
3.4	OS (système d'exploitation)	23
3.4.1	Notre besoin	23
3.4.2	Qu'est-ce qu'un OS ?	23
3.4.3	Liste OS	23
3.4.4	Lequel choisir ?	26
3.4.5	Conclusion	26
3.5	Navigation Web	27
3.5.1	Expression du besoin	27
3.5.2	Technologie existante	27
3.5.3	L'explication de notre choix	28
3.5.4	Conclusion	28
3.6	VPN SSL	29
3.6.1	Le besoin	29
3.6.2	HTTPS	29
3.7	Supervision	29
3.7.1	Msec	29
3.7.2	CheckMyHttps	30
3.7.3	Module test pour clés usb	30
3.8	La défense en profondeur	30
3.9	Guacamole	32
3.10	Complément	32
4	Problèmes rencontrés	35
4.1	Etat du projet PLEIADE à ses débuts	35
4.1.1	Problème	35
4.1.2	Solution	36
4.2	LXC	36

4.2.1	Problème	36
4.2.2	Solution	37
4.3	MMC	37
4.3.1	Problème	38
4.3.2	Solution	38
5	Technologies utilisées	39
5.1	Firewalld	39
5.1.1	Découverte de Firewalld	39
5.1.2	Ces avantages	39
5.1.3	Utilisation de FirewallD	40
5.1.4	Les zones existantes	40
5.1.5	Commandes pour les zones	41
5.1.6	Les services	41
5.2	Msec	42
5.3	Lynis	42
5.4	Guacamole	44
5.4.1	C'est quoi ?	44
5.4.2	Installation	44
5.4.3	Configuration	46
6	Avancement	49
7	La forme finale du projet	51
7.1	Liste des étapes à réaliser	52
7.2	Schéma des étapes	54
8	Conclusion	56
II	Annexes	57
9	Documentation d'installation	58

9.1	Docker	58
9.1.1	Installation de Docker	58
9.1.2	Le fonctionnement de Docker	59
9.1.3	La création d'une image : Utilisation du Dockerfile	59
9.1.4	Utilisation d'une image & lancement d'un conteneur	60
9.1.5	Mise en route de conteneurs	61
9.1.6	Se connecter à un conteneur	61
9.1.7	Commandes utiles	61
9.2	Freelan	62
9.2.1	Packager freelan sur Mageia 6 V1	62
9.2.2	Packager freelan sur Mageia 6 V2	63
9.2.3	Installer freelan avec un package	63
9.2.4	Installer freelan avec le git	64
9.2.5	Configuration freelan	64
9.2.6	Fichier de configuration de Freelan	90
9.2.7	Docker freelan	90
9.3	FirewallD	91
9.3.1	Installation	91
9.3.2	Lancement de Firewalld au démarrage	91
10	Sécurisation Docker	93
10.1	SELinux	94
10.1.1	Approche SELinux pour la sécurisation Linux	94
10.1.2	Les SC (Security Context)	95
10.1.3	Les booléens	95
10.1.4	Les commandes utiles	95
10.2	Portsentry	96
10.3	Fail2ban	97
10.4	rkhunter	97
10.5	Prelude	97
10.6	Surveiller les logs	97

10.7 Côté Linux	98
11 Documentation utilisation utilisateur	99
12 Les tests réalisés	100
12.1 LXC	100
12.1.1 Installation :	100
12.1.2 Tentative en root	101
12.1.3 Tentative en utilisateur lxc	101
12.2 PODMAN	105
12.2.1 Podman vs Docker	105
12.2.2 Tests	106
12.2.3 Conclusion	108
12.3 MMC	108
12.3.1 Test d'installation avec la documentation MMC	108
12.3.2 Test d'installation avec les fichiers MMC ABLogix	110
12.3.3 Test d'installation avec un docker	111
12.3.4 Solution potentielle	112
12.4 FirewallD	112
12.4.1 Nos premiers tests	113
12.4.2 Port forwarding	116
13 Presentation du projet	119
14 Compte-rendu de réunion 1 - 9	121
15 Schémas du projet par ordre chronologique	143
Groupe Epsilon 2018-2019	149
16 Partie Individuelle	149
16.1 BILLETTE Anthony	149
16.2 PESLERBES Jean-Baptiste	150
16.3 PORTIER Théo	151

16.4 RUFFET Simon	151
16.5 YAR Anthony	152
III Complément d'annexes	156
17 Liste des booléens sur SELinux	157
18 MSEC règles et paramètres	173

Première partie

Le projet Pleiade

Pilotage Logiciel d'Équipement Informatique Autonome Distant et Embarqué

Licence

Copyright ou © ou Copr. Bellanger Jean-François, Billette Anthony, Peslerbe Jean-Batiste, Portier Théo, Ruffet Simon, Yar Anthony. En avril 2017.

jean-francois.bellanger@interieur.gouv.fr

Ce logiciel est régi par la licence [CeCILL|CeCILL-B|CeCILL-C] soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence [CeCILL|CeCILL-B|CeCILL-C] telle que diffusée par le CEA, le CNRS et l'INRIA sur le site <http://www.cecill.info>. En contrepartie de l'accessibilité au code source et des droits de copie, de modification et de redistribution accordés par cette licence, il n'est offert aux utilisateurs qu'une garantie limitée. Pour les mêmes raisons, seule une responsabilité restreinte pèse sur l'auteur du programme, le titulaire des droits patrimoniaux et les concédants successifs. A cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs et des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité.

Le fait que vous puissiez accéder à cet en-tête signifie que vous avez pris connaissance de la licence [CeCILL|CeCILL-B|CeCILL-C], et que vous en avez accepté les termes.

Remerciement

Nous remercions Jean-François Bellanger pour avoir initié ce projet. Jean-François a eu une méthodologie différente des autres tuteurs que nous avons eus dans le passé. Nous en avons tiré une bonne expérience dans la gestion de projet. Nous remercions M Aubin qui a su être présent lors des moments de doute dans le projet. Nous remercions Alexandre Babier pour la réalisation du logo. Nous remercions Clémence Peslerbes ainsi que Paul Letourneau qui ont participé à la mise en place de notre maquette en nous prêtant des playmobilis. Il est agréable de travailler avec des personnes motivées et motivantes sur un projet. Nous pouvons aussi nous remercier, pour la contribution et l'implication de chacun dans le projet.

Chapitre 1

Introduction

1.1 La genèse

Durant la présentation des projets scientifiques et techniques (PST) par notre responsable d'année nous avons individuellement accroché à ce projet. C'est par la suite que notre équipe s'est formée. Étant tous issus de la première année à l'ESIEA, nous nous connaissons bien et étions déjà amis. Nous avons déjà réalisé un quart du travail de groupe de ce point de vue.

Le guide des projets scientifiques et techniques stipule : que les groupes doivent être formés de quatre membres maximum. Nous étions cinq. Par conséquent, nous avons fait la demande pour deux sujets : EPSILON et Diode. DIODE étant un module qui sera greffé à PLEIADE. Ce dernier sera chargé d'isoler le réseau interne de consultation du réseau externe internet afin de créer un cercle de confidentialité.

Ensuite nous avons échangé avec notre suiveur, externe à l'ESIEA. Jean-François BEL-LANGER, qui nous a bien fait comprendre lors de notre première réunion que le projet EPSILON et DIODE était trop hétérogène (cf. Rapport de réunion 1 en annexe). Jean-François nous a naturellement proposé deux sujets complémentaires, IGOINE et ESPILON. Le module IGOINE sera chargé d'identifier un utilisateur de confiance. Ces derniers, utiliseront le module PLEIADE après que celui-ci est validé les certificats et clés asymétriques créées par IGOINE. Nous avons découvert que le projet EPSILON s'appuyait sur un projet nommé PLÉIADE. Nous avons donc poursuivi notre étape de compréhension du sujet, par la compréhension du projet PLÉIADE. C'est à ce moment que nous nous sommes rendu compte que le projet PLEIADE est un POC composé d'une constellation de modules décomposés en sous-projet. Et que ces projets possèdent une documentation quasi inexistante. (cf. Rapport de réunion 4 en annexe).

Cet imprévu nous empêche de réaliser les projets IGOINE et ESPILON dans les temps. Nous avons donc décidé de reprendre le sujet PLEIADE et d'y ajouter le projet EPSILON

sans prendre en compte IGOINE. Par conséquent, dans la suite du rapport, nous ne prendrons pas en compte la création des certificats, des clés asymétriques et du périphérique de confiance.

1.2 Notre projet

Le projet est initié par Jean-François Bellanger. Le but est de survenir aux besoins d'accès à distance sur un réseau sécurisé. Il existe déjà un OS qui répond à ce besoin, son nom SPAN (Sécurisation du Poste d'Accès Nomade). SPAN n'est pas multiplateforme. En effet, il permet uniquement de réaliser une connexion distante sur un poste de travail en exploitant les technologies RDP¹ ou VNC². Il existe une autre solution utilisant un système dit de "cage haute" et de "cage basse", son nom ClipOs³. Cette solution nous permet d'accéder depuis le même poste à deux réseaux différents.

Notre projet nous permettra de nous connecter à une plateforme centralisée depuis un système d'exploitation aussi bien Windows, Android ou Linux. Depuis cette plateforme, nous pourrons accéder à différents services qui seront par exemple : la connexion au serveur de fichier ou bien une connexion et une prise de contrôle d'un ordinateur. Nous exploiterons le principe du système dit de "cage haute" et de "cage basse" afin d'isoler notre projet de son hôte qui l'accueille. Le projet EPSILON sera l'édifice de projets annexes qui viendront renforcer la sécurité et la disponibilité de ce service. Nous avons une contrainte forte pour la réalisation de documents pour ce projet afin que ce dernier soit repris plus facilement par nos successeurs. Nous avons réalisé, dans ce but, un état de l'art des différentes technologies qui pourront satisfaire pour le projet (cf. chapitre 3).

1. https://fr.wikipedia.org/wiki/Remote/Desktop_Protocol
2. <https://www.realvnc.com/fr/>
3. <https://clip-os.org/fr/>

Chapitre 2

Gestion de projet

2.1 Notre organisation

En début de projet, lors de la formation de l'équipe, nous avons choisi Anthony Billette comme leader. Son rôle est donc de communiquer par mail et de poser les réunions avec notre suiveur externe et de nous retransmettre les informations que ce dernier lui a transmises. Cette méthode nous permet d'éviter de remplir la boîte mail de notre suiveur, mais surtout de limiter la perte d'information. Ces informations nous sont rapportées via un transfert de mail ou par oral lors des réunions que nous verrons plus bas. Nous avons également séparé le projet en 4 phases. La première est la phase d'étude qui est finie, nous sommes donc passés à la phase de développement. Nous avons également une phase de documentation que nous effectuerons tout au long du projet. Et à la fin nous aurons avoir une phase de test. Afin d'avoir une efficacité optimum, nous avons décidé de nous répartir les tâches. Dans un premier temps, Jean-Baptiste Peslerbes s'occupe de la conteneurisation, Simon Ruffet de la connexion VPN, Anthony Yar des connexions via le firewall, Théo Portier de la MMC et Anthony Billette de la sécurisation via Msec.

Au deuxième semestre, Théo Portier partira à l'étranger. Nous ne serons donc plus que quatre sur le projet. Cependant il effectuera son stage technique de deux mois en janvier-fevrier avec la Police et nous aidera donc à avancer dans le projet.

2.2 Nos Réunions

Dans notre projet, nous participons à trois types de réunions.

Le premier type de réunion se base sur la méthode agile. Cependant, ne travaillant pas à plein temps sur notre projet nous ne sommes pas partis sur des réunions quotidiennes (daily scrum). Nous avons mis en place un système de réunion similaire. En effet, nous effectuons

une réunion a chaque fois que nous effectuons une séance de travail (≥ 3 heures). Dans ces réunions chacun expose ce qu'il a fait depuis la dernière réunion, ce qu'il compte faire durant cette séance de travail et en fin de réunion, notre leader nous donne ou rappelle les informations qu'il a reçues de la part de notre suiveur. Les réunions sont donc plutôt courtes, environ 2 minutes de temps de parole par personne soit entre 10 et 15 minutes. Ces réunions sont importantes puisqu'elles permettent de débloquer les membres du groupe qui sont bloqués, mais surtout de savoir qui travaille sur quoi pour éviter les collisions.

En plus de ces courtes réunions, nous effectuons une réunion toutes les deux à trois semaines avec notre suiveur. Lors de ces réunions, nous lui montrons ce qui a été fait en début de réunion, nous parlons des problèmes rencontrés et cherchons des solutions ensemble. Et, en fin de réunion, nous expliquons ce que nous comptons faire avant la prochaine réunion. Dans les premières réunions, nous avons également essayé de clarifier le projet. Après chacune de ces réunions, nous nous rassemblons afin de débriefer sur cette dernière et nous rédigeons le rapport de réunion. Contrairement aux premières réunions celles-ci sont beaucoup plus longues. Elles durent généralement entre 2 et 3 heures.

Et pour finir, il nous arrive de faire des réunions non officielles avec notre suiveur ESIEA, M. Aubin. Ces dernières nous permettent d'avoir rapidement des réponses à nos questions techniques et de faire part de nos inquiétudes sur telle ou telle technologie. Elles n'ont pas réellement de durée, mais ne dépassent que très rarement les 30 minutes. Elles nous ont permis notamment en début de projet de nous remotiver lorsque nous bloquions et elles permettent également à notre suiveur de voir où nous en sommes.

2.3 Notre salle

Afin d'effectuer notre projet dans des conditions optimales, nous avons demandé un accès à une salle. Cette salle nous permet d'effectuer nos réunions type daily scrum et nos réunions non officielles aisément, mais nous permet également d'installer un serveur et un PC fixe qui nous sert de client. Grâce à cette salle, nous avons accès à un tableau pour faire des schémas et nous avons pu créer notre tableau de sprint avec des post-its (cf tableau post-its).

2.4 Notre Gantt

Notre Gantt décrit des étapes que nous décrivons dans les chapitres suivants. Cf figure Gantt.

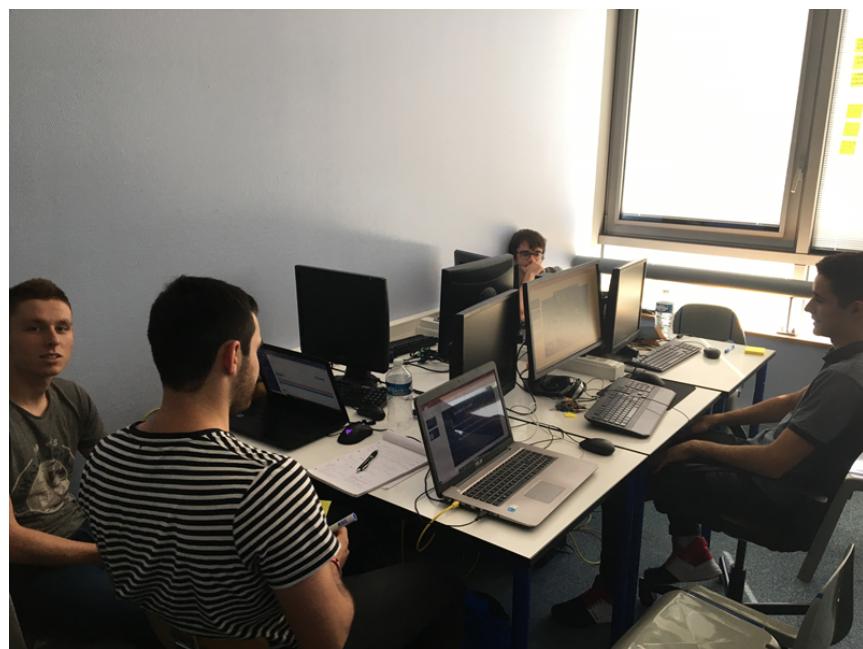


FIGURE 2.1 – Salle

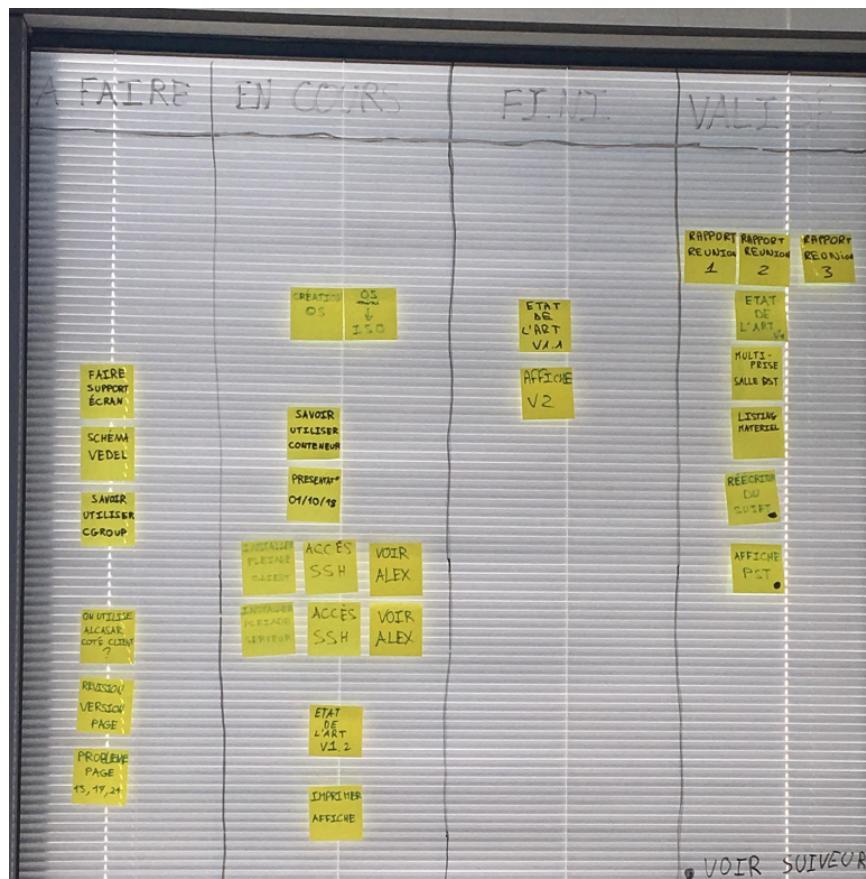


FIGURE 2.2 – Tableau post-its

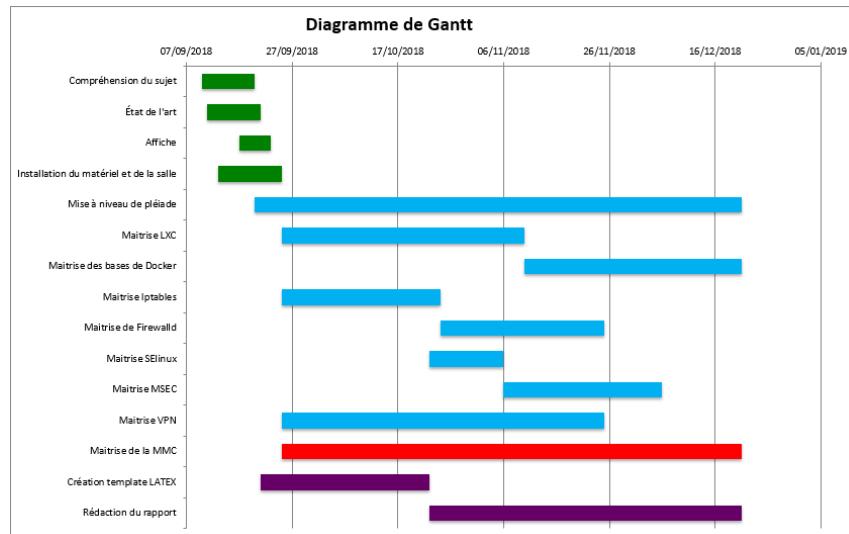


FIGURE 2.3 – Gantt

Chapitre 3

État de l'art

Cet état de l'art a été réalisé au début de projet. Nous n'utiliserons pas les technologies que nous avons validées. En effet, après la réalisation de tests, nous avons dû résoudre des problèmes techniques. Cette état de l'art montre juste les pistes que nous avons pu aborder durant notre PST

Dans cette partie vous trouverez le premier état de l'art que nous avons réalisé pour le projet. Le but était d'acquérir un maximum d'information sur les technologies que nous allons potentiellement utiliser pour le projet. Cet état de l'art n'est pas la version définitive pour le projet.

3.1 VPN

Un élément clef de PLEIADE est la mise en place d'un réseau isolé de l'extérieur, joignable à distance. Ceci impose donc la mise en place d'un VPN (Réseau Privé Virtuel) avec authentification et chiffrement.

3.1.1 Qu'est-ce qu'un VPN ?

VPN est l'acronyme de Virtual Private Network, c'est-à-dire Réseau Privé Virtuel en français. Un VPN est une sorte de tunnel de communication sécurisé à l'intérieur d'un réseau comme internet, permettant de créer un lien direct entre des ordinateurs distants.(Voir figure : Principe du VPN)

Un réseau VPN est tout d'abord basé sur un protocole dit de « tunnelling ». Ce protocole permet de faire circuler les données de manière chiffrée. La connexion entre les ordinateurs est gérée par un logiciel de VPN, créant un tunnel entre eux. C'est le principe de tunnelling qui consiste à créer un chemin virtuel après avoir identifié le destinataire et

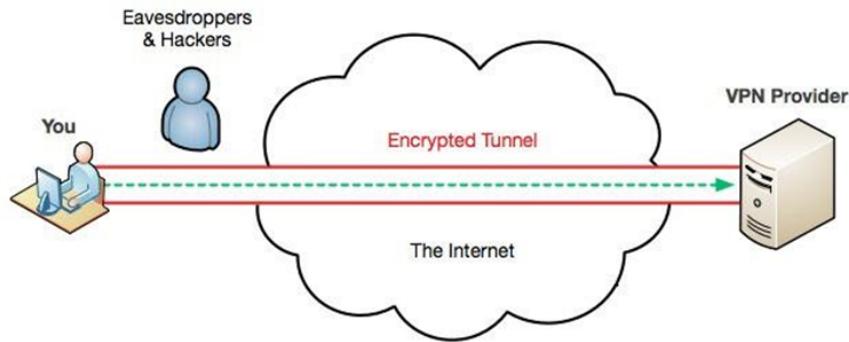


FIGURE 3.1 – Principe du VPN

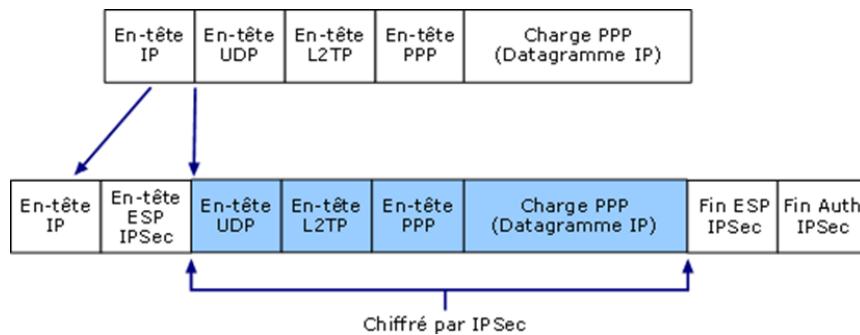


FIGURE 3.2 – Encapsulation

l'émetteur. Par la suite, la source chiffre les données et les envoie en empruntant le chemin virtuel. Les données à transmettre peuvent parfois être prises en charge par un protocole différent. Dans ce cas, le protocole VPN encapsule les données en ajoutant un en-tête. Le tunnelling correspond à l'ensemble des processus d'encapsulation, de transmission et de désencapsulation.(Voir figure : Encapsulation)

3.1.2 Pourquoi utiliser un VPN ?

Un VPN permet une relative protection contre les risques d'espionnages de données sur les réseaux publics ou hotspots. Selon leurs configurations, les chiffrements des VPN peuvent être plus évolués que le HTTPS et permettent d'accéder à internet à travers un tunnel hautement chiffré.

3.1.3 Les différents VPN open-source :

En premier lieu, on peut citer IPSec [6] Ce protocole nécessite un agent sur toutes les machines à mettre en contact pour fonctionner. Il repose avant tout sur une architecture

à clefs privées et une prise de contact avec échange de clef selon l'algorithme d'échange Diffie-Hellman avant de mettre en place une chaîne de communication chiffrée et sécurisée. Il a comme avantage d'opérer sur la couche réseau du modèle OSI et ne nécessite donc pas de modifications des applications pour être exploité. Il intègre aussi un système d'authentification directement dans l'en-tête des paquets, pour garantir une étanchéité des tunnels créés. En revanche, aucun client open source ne supporte le protocole dans son entièreté, et certains routeurs et pare-feu refusent de transférer les paquets IPSec.

Parmi les protocoles open source les plus connus, on peut citer OpenVPN. Celui-ci se base sur OpenSSL pour le chiffrement et l'authentification. Il fonctionne sur un mode client-serveur, il est surtout utilisé comme passerelle VPN. Il fonctionne en créant une interface virtuelle soit TUN (sur la couche réseau) soit TAP (interface Ethernet virtuelle, couche liaison) et en chiffrant toutes les communications au travers de cette interface. Il a comme intérêt d'être très bien implémenté dans les équipements actuels ainsi que fournis avec un ensemble d'outils pour simplifier son intégration et permettre plusieurs modes d'authentification. Mais il ne fonctionne qu'en mode client-serveur ce qui le rend peu modulaire.

Le protocole open source utilisé actuellement est FreelanVPN [1]. Similaire à OpenVPN sur son fonctionnement (suite cryptographique OpenSSL, et adaptateur virtuel TUN/TAP) il diffère en revanche par les différentes topologies de réseau qu'il propose. Il est en effet possible de créer un réseau peer-to-peer (deux clients connectés directement entre eux), une architecture client-serveur ou bien un hybride des deux. Il est également conçu spécifiquement pour créer des réseaux virtuels complets, et non pour agir comme passerelle, bien que cela soit possible. Il est actuellement utilisé dans PLEIADE.

Depuis la création de PLEIADE un nouveau VPN open source a vu le jour : WireGuard. Le principal argument de WireGuard est la simplicité de son code, moins de 4 000 lignes, contre bien plus pour OpenVPN. Pour le spécialiste de la cryptographie Jean-Philippe Aumasson, « WireGuard est plus fiable qu'OpenVPN ». D'une part, par la taille du code. D'autre part, par « le choix des composants cryptographiques, réduisant le risque d'erreur de la part des utilisateurs (on ne peut pas choisir des algorithmes crypto faibles) ». De plus, il sera bientôt intégré au noyau de Linux. Cependant nous pouvons lire sur leur site officiel "WireGuard is not yet complete. You should not rely on this code. It has not undergone proper degrees of security auditing and the protocol is still subject to change. We're working toward a stable 1.0 release, but that time has not yet come.". Wireguard sera donc une piste à explorer pour une future version de PLEIADE.

3.1.4 Conclusion

Pour conclure, WireGuard est sûrement le VPN le plus adapté pour PLEIADE cependant FreelanVPN est déjà utilisé et configuré sur PLEIADE. Le changement du VPN

serait une étape en plus dans notre projet (non prévu dans le sujet) pour finalement avoir le même niveau de sécurité, mais une maintenance du code plus facile. Le temps dépensé par rapport au gain final est trop important.

3.2 Plug-in Firefox

Dans ce projet, nous avons besoin de connecter un OS à une clé USB qui elle-même communiquera avec le serveur. Pour cela nous avons des contraintes, la principale est qu'il nous est impossible de booter sur la clé. C'est pourquoi, nous avons cherché du côté d'un plug-in Firefox. Nous sommes partis sur Firefox car les ordinateurs de la police sont sur Windows et utilisent Firefox.

3.2.1 Qu'est-ce qu'un plug-in ?

Un plug-in est un paquet qui complète un logiciel hôte. Ici c'est donc une extension de Firefox, et cela permet d'ajouter de nouvelles fonctionnalités à l'hôte.

Pour créer un plug-in Mozilla Firefox, nous avons besoin de maîtriser au moins deux langages le XML et le JavaScript. Le premier servira à décrire les données et les interfaces alors que l'autre gérera la relation entre événements et actions.

3.2.2 Quelle est la structure d'un plug-in ?

Un plug-in est en fait un répertoire qui contient obligatoirement un fichier `manifest.json`. Ce dernier contient des métadonnées sur l'extension tels que son nom, sa version, mais également les autorisations qu'elle doit posséder. Ce répertoire contient également des pointeurs vers d'autres fichiers de l'extension, tels que des pages d'arrière-plan, des scripts de contenu, des fichiers d'actions du navigateur (des boutons sur la barre d'outils) ou de la page (boutons sur la barre d'adresse). (Voir la figure : Structure d'un plug-in)

Un des outils qui nous seront sûrement utiles est la possibilité d'utiliser des scripts d'arrière-plan. En effet, nous allons avoir besoin d'effectuer des connexions VPN entre l'utilisateur et le serveur sans que ce soit visible. Ces derniers seront codés dans le fichier `manifest.json`.

HIDAPI est une librairie multi-plate-forme (donc utilisable sur Windows) elle permet une interaction entre une application et une clé USB HID ou un périphérique Bluetooth. Pour utiliser Hidapi dans une application, il suffit qu'un seul fichier source soit déposé dans l'application. (Dernière mise à jour octobre 2017)

La librairie HIDAPI étant ancienne et sa dernière mise à jour datant d'environ 1 an, nous avons décidé de limiter les risques de failles de sécurité en utilisant une autre méthode.

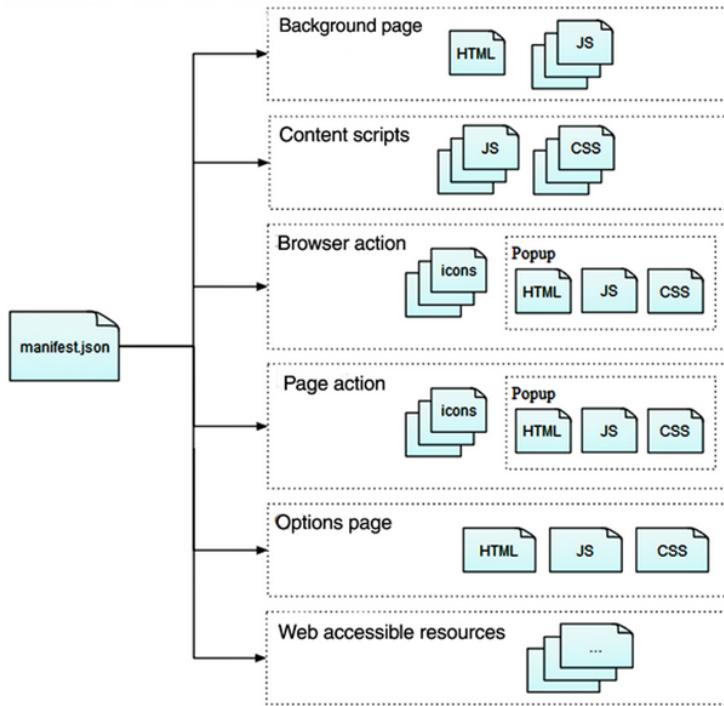


FIGURE 3.3 – Structure d'un plug-in

La deuxième méthode serait de passer par une MMC. La MMC comporte une interface homme-machine codée en PHP. Cette interface fera appel à un ordonnanceur (python) qui gèrera l'authentification et l'intégrité de l'utilisateur. Pour ce faire, elle utilise des scripts Perls. L'un d'entre eux permettra la récupération d'un conteneur LXC présent sur une clé USB, dans le but de monter sur le pc Hôte (Windows). Ce conteneur permettra une connexion au serveur pléiade.

3.2.3 Conclusion

Nous ne sommes pas actuellement fixés sur la technologie à employer.

3.3 La conteneurisation

3.3.1 Les besoins d'EPSILON :

Notre projet a besoin d'un système de virtualisation. En effet, pour plus de simplicité et de sécurité nous voulons cloisonner chaque partie de ce projet pour pouvoir administrer le système plus facilement et le rendre attaquant sur une moins grande surface. Le projet

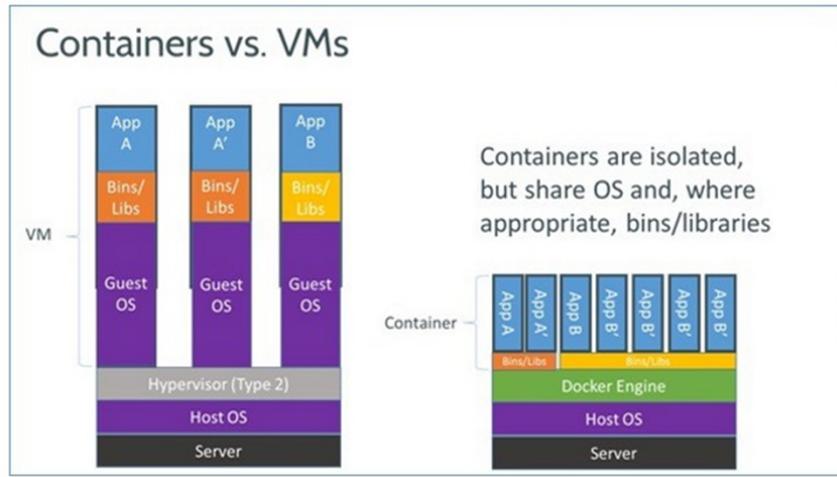


FIGURE 3.4 – Containers vs VMs

doit également être fiable, si une partie du système ne fonctionne plus il est important que ce ne soit que cette partie qui arrête de fonctionner. Nous n'utiliserons qu'un seul OS avec une forte densité d'application. Il faudra donc que le système possède une certaine rapidité et utilise peu de ressources. Enfin, l'idéal serait que ce projet puisse être facilement modulable.

Aux vues des problématiques de notre projet, nous avons choisi la méthode de la conteneurisation.

3.3.2 En quoi ça consiste ?

La conteneurisation est une méthode de virtualisation. C'est-à-dire qu'elle permet de partager les ressources d'un ordinateur physique sur plusieurs machines virtuelles via un hyperviseur. À l'inverse de la virtualisation matérielle, la conteneurisation ne démarre pas d'OS supplémentaire, mais effectue une virtualisation du système d'exploitation. Cela allège considérablement l'utilisation des ressources. Elle offre ainsi à chaque machine un environnement complet d'exécution dans des environnements virtuels isolés que l'on appelle conteneurs.

Autrement dit, chaque conteneur peut être considéré comme une application. (Voir la figure : Containers vs VMs) La conteneurisation offre de nombreux avantages. Tout d'abord, elle est facile à installer, une simple ligne de code suffit. Une fois mise en place, elle est indépendante de la plateforme, donc facilement modulable. De plus, la conteneurisation utilise peu de ressources, ce qui offre une plus grande densité d'applications sur le serveur et évite les pics de virtualisation (le démarrage d'un conteneur s'effectue instantanément). Enfin, la conteneurisation permet l'isolation de chaque application du système. Chaque conteneur

fonctionne donc indépendamment des autres. Ainsi cela facilite grandement l'administration du système et évite le crash complet du système. Cela limite également la surface d'attaque d'un potentiel attaquant.

3.3.3 Les outils de conteneurisation :

LXC « LinuX Containers » : Ce système de virtualisation est la base de la conteneurisation. Il est présent dans le noyau Linux. Grâce à lui, nous pouvons créer une isolation des systèmes de fichier, des identifiants réseau et utilisateur via la création de conteneurs. Il permet également l'isolation des ressources (processeur, mémoire, etc). Son seul point noir est sa gestion. En effet, il n'existe pas à ce jour de solution graphique pour gérer l'organisation des conteneurs. Toutefois, il a récemment été créé un logiciel de gestion de conteneurs en console appelé LXD, qui facilite grandement la gestion des conteneurs LXC.

Docker : Ce projet open source est la solution la plus utilisée dans le domaine de la conteneurisation. Il est basé sur la technologie LXC et y a ajouté de nombreuses capacités. Il offre par exemple la possibilité à l'utilisateur de partager publiquement ses conteneurs (Docker Hub), ce qui rend très facile l'installation d'un conteneur spécifique. La gestion des conteneurs est rendue très simple grâce à l'interface de programmation « Libcontainer » qui démarre, gère et arrête les conteneurs. Docker est disponible sur Linux comme sur Windows.

RKT « rocket » : Cet outil est édité par CoreOS est le concurrent de Docker. Il prend en charge les Images Docker et le format ACI (App Container Images). Les éditeurs se concentrent sur la sécurité (le plus gros point faible de Docker), la compatibilité et une intégration aux standards. Le but étant de fournir les mêmes fonctionnalités que docker et être complémentaire.

Kata Containers est un projet open source et communautaire. Le but de ce projet est d'implémenter des machines virtuelles fonctionnant comme un conteneur (légèreté, performance, modularité...), tout en conservant l'isolation et la sécurité des machines virtuelles. En d'autres termes, il s'agit de faire des conteneurs plus sécurisés et isolés. Kata Containers est encore jeune et il manque donc de la documentation dessus.

Kubernetes est un système de gestion de conteneurs et plus particulièrement un orchestrateur. Il est, à l'origine, un projet Google disponible depuis juin 2014. Il est semblable à tous les autres projets hormis qu'il n'est pas uniquement destiné au Cloud, mais à tout type d'infrastructure. Un de ses points forts étant de permettre de faire de l'Infra As Code (...). De plus, Kubernetes peut gérer un ensemble de technologies de conteneurs qui vont respecter la norme Container Runtime Interface (Docker, RKT, LXD, ...).

3.3.4 Conclusion

Bien que le système Docker soit le plus souvent utilisé par sa simplicité d'utilisation et le fait qu'il offre une interface de visualisation, notre choix s'est finalement porté sur LXD. En effet, notre projet principal (PLEIADE) utilisant LXC, celui-ci nous semblait plus adapté. De plus, de nombreux tutoriels expliquant son utilisation sont disponibles et permettent de comprendre son utilisation. En outre, le projet "conteneur box" créé par Alexandre DEY permet l'automatisation de la création de conteneurs LXC, ce qui nous permet de gagner un temps considérable.

C'est pourquoi nous utiliserons LXC dans la suite de notre projet.

3.4 OS (système d'exploitation)

3.4.1 Notre besoin

Notre projet demande de réaliser un environnement sécurisé afin d'accéder à un espace de travail à distance à partir d'un ordinateur quelconque. La solution retenue à l'heure actuelle est de faire tourner un OS minimal à l'aide de multiples conteneurs afin de minimiser les risques d'infection du système.

3.4.2 Qu'est-ce qu'un OS ?

Un système d'exploitation (souvent appelé OS — de l'anglais Operating System) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs. Il reçoit des demandes d'utilisation des ressources de l'ordinateur — ressources de stockage des mémoires (par exemple des accès à la mémoire vive, aux disques durs), ressources de calcul du processeur central, ressources de communication vers des périphériques (pour parfois demander des ressources de calcul au GPU par exemple ou toute autre carte d'extension) ou via le réseau — de la part des logiciels applicatifs. Le système d'exploitation gère les demandes ainsi que les ressources nécessaires, évitant les interférences entre les logiciels.

3.4.3 Liste OS

Notre projet nous contraint à utiliser que des OS étant compatibles avec des conteneurs (LXC, Docker, ...), mais aussi assez légers, car nous souhaitons l'installer sur un périphérique nomade. C'est aussi pour cela que nous devons faire attention au fait qu'il doit être facilement installable sur un périphérique nomade.

Voici une liste non exhaustive d'OS potentiels :

Fedora

Fedora : est un système d'exploitation libre et une distribution GNU/Linux communautaire développée par le projet Fedora et sponsorisée par l'entreprise Red Hat, qui lui fournit des développeurs ainsi que des moyens financiers et logistiques.

Famille : GNU/Linux

Langues : Multilingue

Type de noyau : Noyau Linux

État du projet : Développement actif

Entreprise / Développeur : Fedora Project

Licence : Licence libre

Première version : Novembre 2003

Dernière version stable : 28

Dernière version avancée : 29

Méthode de mise à jour : DNF

Environnement de bureau : GNOME, KDE, Xfce, LXDE, MATE, LXQt, Cinnamon et Sugar

Gestionnaire de paquets : RPM Package Manager

Site web : getfedora.org

Mageia

Mageia : Mageia est un système d'exploitation libre, basé sur GNU/Linux. C'est un projet communautaire, soutenu par une association loi 1901 constituée de contributeurs élus.

Famille : Type Unix

Langues : Multilingue

Type de noyau : Monolithique modulaire

État du projet : Développement actif

Estat des sources : Logiciel libre

Entreprise / Développeur : Fedora Project

Licence : Diverses

Première version : 1er juin 2011

Dernière version stable : 6.0

Méthode de mise à jour : urpmi

Environnement de bureau : KDE, GNOME, XFCE, LXDE, Cinnamon, Mate et d'autres

Gestionnaire de paquets : RPM Package Manager

Site web : mageia.org

Mandriva Linux

Mandriva Linux : (Anciennement Mandrakelinux) est un système d'exploitation développé par l'entreprise Mandriva de 1998 à 2012. Ciblant à la fois le grand public et les professionnels, cette distribution GNU/Linux est construite autour de l'environnement graphique KDE.

Famille : GNU/Linux

Langues : Multilingue

Entreprise : Mandriva SA

État du projet : Arrêté

Etat des sources : Logiciel libre

Entreprise / Développeur : Fedora Project

Licence : Diverses

Première version : 23 juin 1998

Dernière version stable : 2011.0 (Hydrogen)

Dernière version avancée : 27 juillet 2011

Méthode de mise à jour : urpmi

Environnement de bureau : KDE

Site web : www.mandriva.com

Alpine linux

Alpine Linux : est une distribution Linux ultra-légère, orientée sécurité et basée sur Musl et BusyBox, principalement conçue pour un « utilisateur intensif qui apprécie la sécurité, la simplicité et l'efficacité des ressources ». Elle utilise les patchs PaX et Grsecurity du noyau par défaut et compile tous les binaires de l'espace utilisateur et exécutables indépendants de la position (dits “portables”) avec protection de destruction de la pile.

Famille : GNU/Linux

Langues : Multilingue

Entreprise : Mandriva SA

État du projet : Actif

Etat des sources : Logiciel libre

Entreprise / Développeur : Alpine Linux development team

Licence : Opensource

Dernière version stable : 3.8.0

Méthode de mise à jour : APK

Environnement de bureau : BusyBox

Gestionnaire de paquets : Alpine package manager

Site web : <https://alpinelinux.org>

3.4.4 Lequel choisir ?

Fedora : peut-être un excellent choix, le système comporte déjà des packages de gestion des conteneurs nativement. Peu pratique pour le modularisé comparé à Mageia, ne comporte pas de package “alcasar” dans ses dépôts.

Alpine OS : Système trop léger, ne prend pas en compte les containers nativement, de plus il est plus difficile à tenir à jour.

Mageia : le système devant être choisi finalement, comporte “alcasar” dans ses dépôts, la prise en charge des conteneurs est gérée maintenant dans les nouveaux dépôts (LXD, LXC, DOCKER, … sont maintenant présents).

Mandriva Linux : OS délaissé depuis 2011, remplacé par Open-Mandriva, reste une bonne alternative à Mageia.

3.4.5 Conclusion

Nous utiliserons la distribution Mageia. C'est tout d'abord une contrainte donnée par notre suiveur mais aussi pour plusieurs raisons de sécurité. Tout d'abord Mageia est un OS qui a nativement tous les packages le comprenant qui ne comporte aucun warning, donc ils sont plus sécurisés, Mageia a également une énorme communauté derrière le développement de l'OS et de ses packages en cas de soucis nous savons que nous pourrons trouver une personne pouvant nous épauler.

Page	Site	Format	Moteur de rendu	HTML5	Interface	Description
Firefox			Gecko	474/555	Bonne intégration à Unity .	Navigateur web multiplateforme de référence, libre, extensible, personnalisable, avec des performances inégalées. C'est le navigateur par défaut de la plupart des variantes d'Ubuntu.
Chromium			Blink	516/555	intégration GTK+ correcte	Navigateur open-source de Google, servant de base à Google Chrome . Navigateur par défaut d' Ubuntu Budgie . Il est compatible avec les extensions du Chrome Web Store.
QupZilla			QtWebEngine (basé sur WebKit)	320/555	Qt	Navigateur extrêmement léger ²¹ basé sur QtWebEngine (qui remplace QtWebKit). Très adapté à KDE ou LXQt , il est très portable (et peut même être utilisé sur Windows). Le développement de QupZilla a été abandonné au profit de Falkon .
Falkon			QtWebEngine (basé sur WebKit)	non testé	Qt	Successeur de QupZilla , intégré aux projets KDE .
GNOME Web			WebKitGTK+	397/555	GTK+ 3	Navigateur web épuré du projet GNOME , intégré et adapté à l' ergonomie de GNOME Shell .
Eolie			WebKitGTK+	397/555	GTK+ 3	Navigateur épuré et léger, mais complet et innovant, particulièrement adapté à un environnement GNOME 3 .
Midori			WebKitGTK+	319/555	GTK+ 3	Navigateur ultra-léger récemment intégré au projet Xfce. C'est aussi le navigateur par défaut dans Elementary OS .
Vivaldi			Blink	516/555	Thème très personnalisable	Navigateur multi-plateforme, basé sur Chromium , qui a pour but d'implémenter les fonctionnalités avancées d' Opera 12 . Il est compatible avec les extensions du Chrome Web Store.
Konqueror			KHTML	487/555	Qt	Couteau suisse du bureau KDE Plasma , c'est entre autres un navigateur web très moderne et performant, et un gestionnaire de fichiers riche en fonctionnalités pour KDE.
Min			Electron	516/555		Navigateur simple, léger et multiplateforme, Min est développé avec la technologie Electron.

FIGURE 3.5 – Screenshot de la documentation Ubuntu 1/2

3.5 Navigation Web

3.5.1 Expression du besoin

Nous avons besoin d'un navigateur web léger, rapide et surtout sécurisé. Le navigateur doit s'intégrer à un environnement Mageia car l'OS Mageia est une contrainte forte de notre projet. Le navigateur web servira d'interface graphique à notre distribution Mageia. Le but est de pouvoir se connecter à un serveur distant depuis cette interface web (navigateur web). Nous avons besoin d'avoir un navigateur qui sera mis à jour dans le temps et par conséquence fiable.

3.5.2 Technologie existante

Document du site web <https://doc.ubuntu-fr.org/navigateur> comparant les différents navigateurs.(Voir figure : Screenshot de la documentation Ubuntu). Source : <https://doc.ubuntu-fr.org/navigateur> consulté le : 14/09/18

Brave			Blink	526/555	intégration GTK+ correcte	Navigateur open-source basé sur Chromium, mais plus rapide.
Pale Moon			Goanna	397/555		Navigateur léger, rapide et personnalisable, basé sur une ancienne version de Firefox.
Opera			Blink	518/555		Navigateur gratuit, rapide extensible et sécurisé, basé sur Chromium.
Navigateur Web Ubuntu			Blink	516/555	Ubuntu Phone	Le "Navigateur Web Ubuntu" (webbrowser-app), basé sur Chromium et développé par Canonical pour la convergence desktop, smartphone et tablette, est le navigateur de la tablette BQ Aquaris M10 vendue avec Ubuntu.
Iridium browser			Blink	516/555	intégration GTK+ correcte	Basé sur Chromium, un navigateur stable et open-source, "dégooglisé" et sans intrusion dans votre vie privée. Hélas, il n'est pas disponible en français pour Ubuntu
Google Chrome			Blink	518/555	intégration GTK+ correcte	Version non-libre de Chromium, avec des améliorations mais davantage d'immiscions dans votre vie privée.
TOR browser			Gecko	362/555		Navigateur basé sur Firefox, qui protège la vie privée, et permet de surfer anonymement sous certaines conditions. Il permet aussi d'accéder au réseau anonyme W TOR .
Slimjet			Blink	516/555	intégration GTK+ correcte	Navigateur (propriétaire ?) basé sur Chromium avec des fonctionnalités pré-intégrées comme le blocage de publicités et le téléchargement de vidéos depuis le net. Télécharger le fichier deb à partir de la page téléchargement du site officiel
Rekonq			Webkit	279/555	Qt	Un navigateur ultra-léger, bien intégré à KDE, mais dont le développement s'est arrêté en 2014 ³⁰ .
uzbl			Webkit	308/555		Navigateur graphique très léger, avec interface en lignes de commande.

FIGURE 3.6 – Screenshot de la documentation Ubuntu 2/2

3.5.3 L'explication de notre choix

Dans un premier temps, nous nous sommes tournés vers le navigateur Midori, simple, léger et efficace, c'était le meilleur candidat pour notre besoin. Mais après quelques recherches, nous nous sommes rendu compte que Midori n'est plus mise à jour depuis 2 ans. Puis dans un second temps, nous nous sommes demandé quel était le navigateur le plus fiable dans 5 ans ? nous nous sommes naturellement tourné vers le navigateur Firefox, la référence en termes de navigateur web libre. En effet, ce navigateur a une énorme communauté qui le maintient en condition opérationnelle. De plus, Firefox est simple d'installation sur Mageia et reste la référence en termes de rapidité.

3.5.4 Conclusion

Nous utiliserons Firefox comme interface web/graphique pour notre os. Nous l'utiliserons en mode Kiosk. Nous n'avons pas encore défini la technologie que nous allons utiliser à ce niveau. R-Kiosk paraissait être un bon candidat, mais ce dernier n'est plus disponible sur le portail des plug-ins Firefox.

3.6 VPN SSL

3.6.1 Le besoin

Nous allons utiliser un VPN SSL pour utiliser le HTTPS. Cela permettra d'améliorer la sécurité pour la communication entre les clients et le serveur.

3.6.2 HTTPS

HTTPS permet au visiteur de vérifier l'identité du site web auquel il accède, grâce à un certificat d'authentification émis par une autorité tierce, réputée fiable (et faisant généralement partie de la liste blanche des navigateurs internet). Il garantit théoriquement la confidentialité et l'intégrité des données envoyées par l'utilisateur (notamment des informations entrées dans les formulaires) et reçues du serveur. Il peut permettre de valider l'identité du visiteur, si celui-ci utilise également un certificat d'authentification client.

3.7 Supervision

La supervision est caractérisée par tout élément passif de notre projet qui permettra de suivre les activités d'un système que nous qualifierons de sensible pour la sécurité de notre projet.

3.7.1 Msec

MSec (Mandriva Security)¹ est une application développée par Mandriva. MSec agit directement sur la sécurité du système d'exploitation de la famille de Mandriva. Dans notre cas, nous utilisons Mageia qui est un fork de l'équipe de développement de chez Mandriva dans la création d'un nouveau système d'exploitation. MSec permet de connaître les modifications (écriture ou changement de droit) sur les fichiers et dossiers que l'on souhaite surveiller. Nous n'avons pas encore vérifié, mais il semblerait que MSec est capable de détecter un fichier qui a été changé par son fichier initial. C'est très utile quand un attaquant veut modifier un fichier de configuration système et par conséquent prendre directement ou indirectement le contrôle sur la machine.

Msec pourra détecter cette modification et remettre le système dans un état que nous aurions fixé au préalable.

1. <https://wiki.mageia.org/en/Msec>

3.7.2 CheckMyHttps

CheckMyHttps "une méthode simple pour vérifier que nos connexions web sécurisées (protocole HTTPS) ne sont pas interceptées (ni déchiffrées, ni écoutées, ni modifiées). Cette méthode exploite un module additionnel (appelé "extension") pour votre navigateur web." Ce plugin sera ajouté à notre navigateur Firefox afin de garantir les propriétés ACID² de notre connexion. Ce plugin est développé par M. Rey chercheur au laboratoire CVO de L'ESIEA.

3.7.3 Module test pour clés usb

Le module de test pour la clé USB a été élaboré par Hippolyte Bernard et Alexandre DEY lors de la première version de PLEIADE. Il permet de monter la clé USB sur un container en s'assurant qu'elle ne possède pas de virus. Le cas échéant, l'utilisateur est averti que sa clé possède un virus ou qu'elle ne peut pas être montée (exemple : faute de droits). Ce module va être utilisé par notre projet. Il n'a pas encore été testé mais le sera lors du second semestre.

3.8 La défense en profondeur

Le principe de Vauban (ou défense en profondeur) est un principe fort en sécurité informatique qui est de plus en plus utilisé. Le principe est de réaliser plusieurs niveaux de défense différents afin de diminuer les failles potentielles dues à l'utilisation d'un unique moyen de défense (Un seul mur de défense pour l'exemple des murs défensifs d'un château fort).

En informatique le principe de Vauban, peut être réalisé sur plusieurs points : -Les équipements (proxy, sgbd, pare-feu dynamique, pare-feu statique, ...) -Les solutions logiciels (Les langages de développements, les logiciels de sécurité) -Les solutions réseaux (Cloisonnement par Vlan, DMZ, Diode, ...).

Un exemple est montré par une architecture 3 tiers pour une application web sur la figure 3.7

Dans le cadre de notre projet, le principe de Vauban est au cœur de la sécurité. En effet nous allons utiliser la rupture protocolaire afin de protéger au mieux les données des utilisateurs. Les points utilisant ce principe sont : La MMC (L'architecture 3 tiers, un langage différent pour l'affichage, l'ordonnanceur et, les scripts), l'encapsulation des différentes couches du projet. Le schéma Figure 3.8 montre la profondeur de notre défense sur notre application.

2. https://fr.wikipedia.org/wiki/Propriétés_ACID

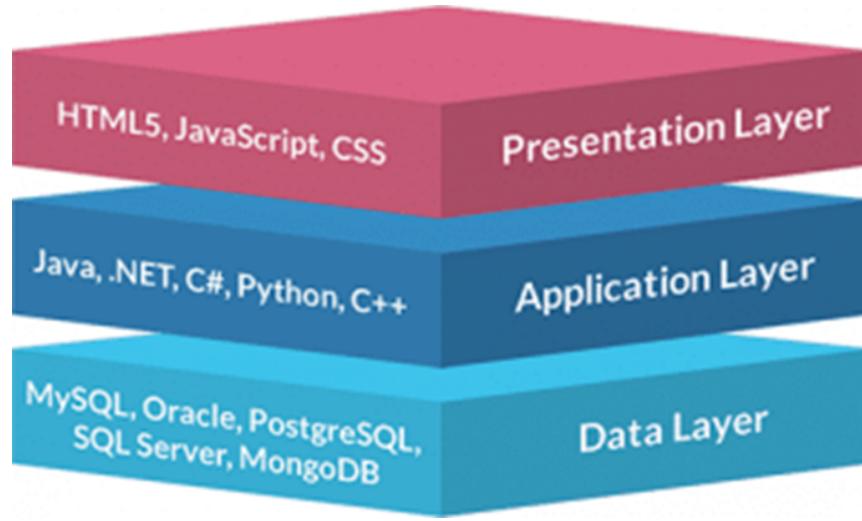


FIGURE 3.7 – Architecture 3 tiers pour une application web

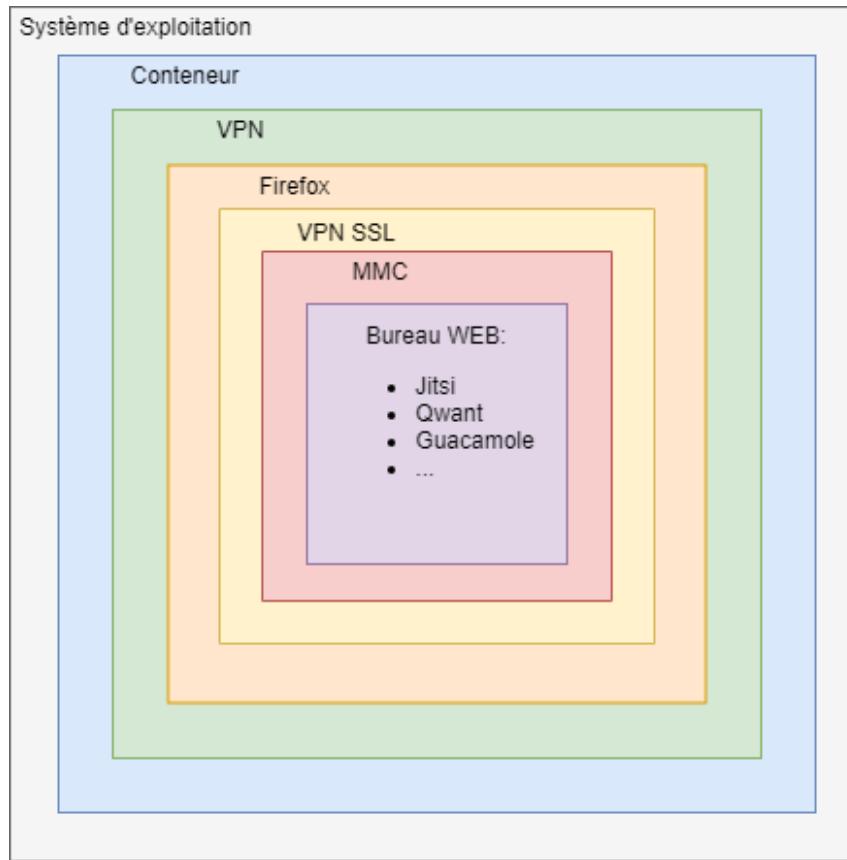


FIGURE 3.8 – Schéma de la défense en profondeur pour le projet EPSILON

3.9 Guacamole

Apache guacamole (communément appeler guacamole) est une application server permettant à l'utilisateur de communiquer en SSH, VNC ou RDP avec un autre ordinateur. C'est une application dite « Clientless » car guacamole ne nécessite d'aucune installation pour qu'un client l'utilise. Dans notre projet, guacamole sera sous forme d'un plugin sur le serveur. Ainsi le client pourra l'utiliser dans notre outil facilement en le sélectionnant lors de la connexion à la MMC.

3.10 Complément

Utilisation de Gecko et Xming

Notre besoin :

Maintenant que nous avons choisi le navigateur web il nous faut un environnement graphique le plus minimaliste possible. Nous avons donc fait des recherches du côté de Xming, Fluxbox et de Gecko.

Xming :

Xming est un portage sous Windows du système de fenêtrage X ouvert des systèmes Unix et Linux. Il est basé sur le serveur X.org et est compilé par MinGW. Il est utilisé afin de rediriger l'affichage vers Windows d'une application graphique. Dans notre projet, il nous permettra d'afficher un Firefox lancé via la couche Ubuntu de Windows 10. (Cause un problème de sécurité si l'affichage utilise Windows?)

Fluxbox :

Pour la partie 100% Linux de notre projet nous avons cherché du côté de Fluxbox. Fluxbox est un gestionnaire de fenêtres très léger, il est notamment utilisé dans le but de relancer des vieux ordinateurs qui ne supporteraient pas des environnements graphiques comme GNOME ou KDE. L'un des avantages de Fluxbox est qu'il est entièrement personnalisable. Il est donc aisément d'enlever toutes les fonctions qui ne sont pas nécessaires au projet, tels que les raccourcis clavier ou encore les raccourcis de navigation comme les clics droits. Fluxbox étant un environnement très léger la possibilité de rencontrer des failles de sécurité est réduite. Lors de nos tests, nous avons également réussi à lancer Firefox en mode Kiosque au démarrage de Fluxbox.

Gecko :

Après avoir réussi à lancer Firefox en mode kiosque nous nous sommes demandé si Firefox ne possédait pas trop de fonctionnalité pour son utilité dans le projet. C'est pourquoi nous avons fait des recherches sur Gecko qui est un logiciel moteur de rendu pour présenter des pages web. Il a été lancé par Mozilla en 1998, il est open source et libre. Aujourd'hui,

il est utilisé par des applications telles que Firefox ou encore Thunderbird. Gecko possède le minimum nécessaire à l'affichage de page web sur un écran. Il respecte les standards du web et les recommandations W3C*. De plus, il fonctionne sur les systèmes d'exploitation Windows, Linux et Mac OS, ce qui est très intéressant pour nos deux parties du projet.

Définitions : IPsec (Internet Protocol Security), défini par l'IETF comme un cadre de standards ouverts pour assurer des communications privées et protégées sur des réseaux IP, par l'utilisation des services de sécurité cryptographiques, est un ensemble de protocoles utilisant des algorithmes permettant le transport de données sécurisées sur un réseau IP. IPsec se différencie des standards de sécurité antérieurs en n'étant pas limité à une seule méthode d'authentification ou d'algorithme et c'est la raison pour laquelle il est considéré comme un cadre de standards ouverts. De plus, IPsec opère à la couche réseau (couche 3 du modèle OSI) contrairement aux standards antérieurs qui opéraient à la couche application (couche 7 du modèle OSI), ce qui le rend indépendant des applications, et veut dire que les utilisateurs n'ont pas besoin de configurer chaque application aux standards IPsec.

TLS (ou SSL) est un protocole de sécurisation d'échange sur internet fonctionnant suivant un mode client-serveur. Il permet de satisfaire aux objectifs de sécurité suivants :

- l'authentification du serveur ;
- la confidentialité des données échangées (ou session chiffrée) ;
- intégrité des données échangées ;
- De manière optionnelle, l'authentification du client (mais dans la réalité celle-ci est souvent assurée par le serveur).

NAC Un contrôleur d'accès au réseau est une méthode informatique permettant de soumettre l'accès à un réseau d'entreprise grâce à un protocole d'identification de l'utilisateur et au respect par la machine de cet utilisateur des restrictions d'usage définies pour ce réseau. Architecture ARM C'est un style d'architecture processeurs. TUN/TAP est une fonction de réception et de transmission de paquets entre le noyau et les programmes de l'espace utilisateur. Cette fonction peut être vue comme une simple interface point à point ou Ethernet qui, au lieu de recevoir les paquets d'un média physique, les reçoit du programme de l'espace utilisateur. De même, cette interface au lieu d'envoyer les paquets vers un média physique, les transmet au programme de l'espace utilisateur.

MMC : Mageia Management Console W3C : World Wide Web Consortium est un organisme de standardisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web telles que HTML5, HTML, XML, RDF, SPARQL, CSS, XSL, PNG, SVG et SOAP.

Outils Alcasar : ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau) est un contrôleur d'accès au réseau (Network Access Controller - NAC) qui trace, impute, et protège les connexions à Internet. Ce projet libre et gratuit permet aux responsables de réseaux de consultation de répondre aux exigences de la loi

française (LCEN). Il intègre un système de filtrage dynamique par utilisateur composé d'un antimalware et d'une liste de noms de domaine et d'adresses IP (liste de l'université de Toulouse). Cette liste est exploitable en mode "liste noire" (blacklist) ou en mode "liste blanche" (whitelist)

Auteurs : Jean-Philippe (JP) Aumasson is Principal Research Engineer at Kudelski Security, in Switzerland. He obtained his PhD in cryptography from EPFL in 2010. JP designed the popular cryptographic functions BLAKE2 and SipHash, and the authenticated cipher NORX. He presented at Black Hat, DEFCON, RSA, and other international conferences. He initiated the Crypto Coding Standard and the Password Hashing Competition projects, and wrote the books *The Hash Function BLAKE* (Springer, 2015) and *Serious Cryptography* (No Starch Press, 2017).

Chapitre 4

Problèmes rencontrés

4.1 Etat du projet PLEIADE à ses débuts

4.1.1 Problème

Lorsque nous avons commencé notre état de l'art nous sommes partis du principe que PLEIADE était un POC fonctionnel auquel nous devions rajouter le module EPSILON (le sujet de notre PST). Nous n'avons donc pas fait de recherches approfondies sur celui-ci. Cependant après avoir réalisé notre état de l'art, Simon et Jean-Baptiste ont voulu installer la partie cliente et la partie serveur de PLEIADE. Au vu d'un manque important de documentation, des échanges par mail ont été réalisés avec l'initiateur du projet Alexandre DEY. Malgré cela le projet restait flou et ils n'arrivaient pas à installer PLEIADE.

Alexandre DEY a donc gentiment accepté une réunion à son domicile pour pouvoir nous aider (compte-rendu numéro 4 en annexes). Nous nous sommes alors rendu compte de l'avancée réelle de PLEIADE. En effet, nous avons découvert que le projet n'était pas opérationnel. Il manquait au projet de la documentation et des scripts d'installation et de lancement de PLEIADE. De plus, certains bugs n'avaient pas été résolus pendant la création de ce POC et celui-ci avait été développé pour la distribution Fedora alors que nous devions utiliser Mageia. Ainsi il n'était pas possible d'installer PLEIADE en l'état et la compréhension du travail déjà réalisé devait passer par du reverse-engineering.

Il en a résulté que nous ne pouvions pas créer le module complémentaire EPSILON sans avoir PLEIADE fonctionnel.

4.1.2 Solution

Après cette découverte, nous avons expliqué la situation à nos deux suiveurs en leurs disant que les objectifs qui avaient été fixés n'étaient plus atteignables. Nous avons donc réalisé une refonte des objectifs durant une réunion avec notre suiveur (compte-rendu numéro 5 en annexe). Il a été convenu que nous devions commencer par rendre PLEIADE fonctionnel. Deux choix se présentaient :

- Comprendre le code d'Alexandre DEY, adapter les technologies, changer de distribution et réaliser la documentation.
- Repartir de zéro en s'inspirant des choix et des codes qu'Alexandre DEY avait réalisés.

La deuxième option a été choisie. L'objectif du premier semestre a donc été de refaire PLEIADE sur mageia avec de la documentation et fonctionnel. L'objectif du semestre deux est de finir PLEIADE et de commencer le module EPSILON si le temps le permet.

4.2 LXC

4.2.1 Problème

Suite à notre état de l'art, nous sommes tombés sur la conclusion que la solution de conteneurisation LXC (Linux Containers) était la plus adaptée à notre projet. Cependant, lors de tests, nous avons rencontré plusieurs problèmes qui nous ont fait comprendre que son utilisation allait être compliquée.

Premièrement, sous la distribution du projet (mageia), il n'y a pas ou très peu de documentation sur l'installation et l'utilisation de LXC. Cela a rendu la tâche difficile dès le début. En effet, il fallait adapter toutes les documentations à notre distribution. N'ayant pas les connaissances parfaites concernant les deux OS, cela s'est vite avéré fastidieux et bloquant par moment.

Puis mageia n'a pas la configuration par défaut pour créer des conteneurs en mode non privilégié. En effet, certains fichiers normalement présents automatiquement sur d'autres distributions ne le sont pas sur LXC. Il faut donc les créer, comprendre leurs fonctionnements, paramétrier les fichiers de configuration et cela prend du temps.

Un autre point noir est que ce logiciel ne possède pas d'image pour la distribution mageia. Nous ne pouvions donc pas créer un conteneur possédant un OS mageia. Nous nous sommes donc renseignés auprès des développeurs de mageia pour en créer une. C'est possible mais beaucoup trop long pour le temps qui nous était imparti.

Enfin EPSILON devait aussi pouvoir être mis sur Windows. Cependant, LXC ne fonctionne que sur un linux. On devait donc installer une couche Linux à un Windows afin que

cela puisse fonctionner. De plus, cela nous obligeait à faire des scripts d'installation pour Linux et pour Windows.

Pour conclure, nous avions beaucoup de problèmes. Aucun n'était bloquant mais tous prenaient du temps à résoudre, et nous ne disposions pas de ce temps là.

4.2.2 Solution

Pour pallier les nombreux problèmes, nous avons choisi d'opter pour une autre solution de conteneurisation : docker.

Celle-ci est bien documentée, simple d'installation et d'utilisation. Ses commandes sont très explicites.

De plus, elle permet au projet d'être facilement maniable. En effet, docker s'installe sur un Windows comme sur un Linux. Il n'est donc pas nécessaire de faire des scripts différents pour installer et configurer docker. Cela nous facilite donc grandement la tâche. Au cas où le projet EPSILON voudrait changer de distribution, il n'y aurait aucun souci.

Enfin, elle est très utilisée dans le monde et régulièrement mise à jour. Cela nous permettra ainsi de corriger des failles de sécurité régulièrement.

En conclusion, nous allons utiliser la solution de conteneurisation docker.

4.3 MMC

La Mandriva Management Console (ou MMC) est une application web modulable permettant de gérer des parcs informatiques facilement. Elle est modulable du fait de sa structure. En effet la MMC basique n'est qu'une coquille vide où il faut rajouter des éléments pour la voir fonctionner et lui donner de nombreuses possibilités différentes à l'aide de plug-in. C'est alors que d'une simple coquille vide nous pouvons obtenir un outil permettant de gérer de nombreux ordinateurs avec la possibilité de déploiement rapide d'OS ou d'application sur des ordinateurs distants.

L'avantage de la MMC au niveau sécurité est la manière dont elle est construite, en effet le fonctionnement est divisé en 3 parties distinctes qui sont : L'interface web (Le front-end), l'ordonnanceur (Le MMC agent), et les scripts(le Back-end, les services de la MMC)

le schéma figure 4.1 montre ce fonctionnement.

Nous utilisons une MMC dans notre projet afin de gérer les utilisateurs de notre application ainsi que le déploiement de logiciel spécifique sur les ordinateurs des utilisateurs. Le premier module que nous souhaitons ajouter est le plug-in "guacamole".

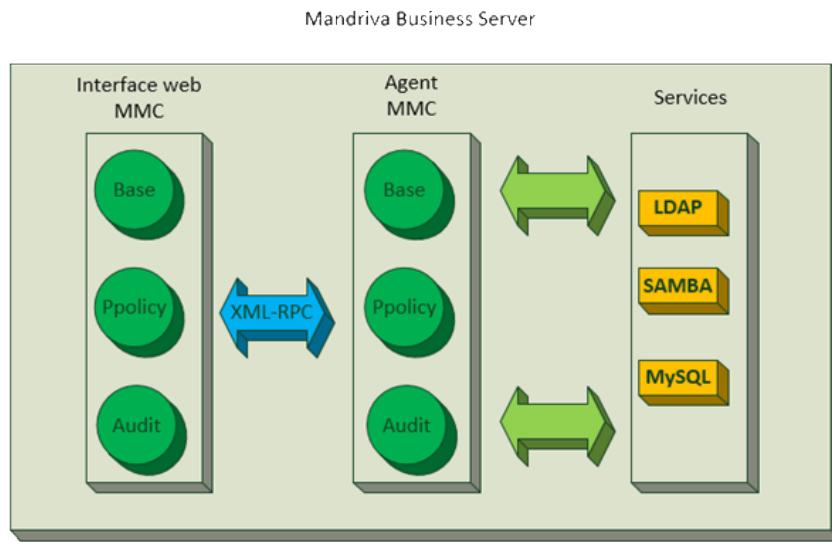


FIGURE 4.1 – Schéma de la MMC

4.3.1 Problème

Nous avons rencontré de nombreux problèmes ensemble pour le moment qui sont des soucis de paquets sur la MMC, ainsi que du système d'exploitation que nous souhaitions utiliser.

Tout d'abord, la MMC à l'heure actuelle est maintenue par l'entreprise "SIVEO" qui a développé un module pour la MMC appelé "Pulse2". Les paquets de base de la MMC installent donc les plug-ins liés à leur module ce qui permet d'avoir un outil d'ores et déjà complet pour commencer à l'utiliser. Cependant les paquets sont à jour uniquement sur *Debian*, ils ont été maintenus sur *Mageia* mais ne sont pas fonctionnels. Lors de l'installation des paquets, le script étant sensé lancer l'agent de la MMC plante et ne s'installe pas correctement, ce qui empêche la MMC de fonctionner. Cela a été testé par notre groupe. La procédure des différents tests est présente en annexe 10.3 .

4.3.2 Solution

Les solutions pour l'instant sont de contacter le mainteneur des paquets afin qu'il règle le souci d'installation et de travailler avec lui afin de réaliser une MMC fonctionnelle dans les plus brefs délais. Les solutions de changement de système d'exploitation ne sont pas pris en compte car nous devons être sur *Mageia* afin de conserver un outil stable au fur et à mesure des mises à jour des paquets étant utilisés dans la MMC. L'utilisation d'un outil autre que la MMC est aussi trop contraignant au vu de la puissance de la MMC quand à sa modularité et sa stabilité.

Chapitre 5

Technologies utilisées

5.1 Firewalld

5.1.1 Découverte de Firewalld

Dans un premier temps, nous sommes partis sur Iptables puisque c'est installé de base dans les distributions Linux. Cependant, en faisant les tests, nous nous sommes aperçus que toutes les règles Iptables étaient supprimées au démarrage. Pour remédier à ce problème, nous avons créé un script que nous lancions au démarrage. Cette fois les règles étaient visibles, mais non actives. Nous avons donc regardé dans le journalctl et nous avons aperçu que Shorewall était présent et qu'il passait après le script et prenait le contrôle des Iptables. Nous avons donc fait des recherches, sur Shorewall et avons découvert l'existence de Firewalld. D'après nos recherches, ce dernier est plus simple à paramétriser que Iptables et possède de nombreux avantages.

5.1.2 Ces avantages

Contrairement à Iptables et Shorewall, il est dynamique, ce qui veut dire qu'il est possible de changer les règles sans redémarrer la machine ou le service.

Il possède une interface Firewalld D-bus, qui simplifie l'adaptation des paramètres des services et des applications.

Il possède une option "-permanent" qui permet de garder les règles même après un redémarrage et nous évite l'écriture d'un script. Il nous est également possible de ne pas l'utiliser afin de faire des tests en live.

Il possède un système de zones dans lesquelles il est possible de définir des paramètres, autoriser / interdire des services ou encore autoriser / interdire des ports.

De plus, il est utilisé dans Alcasar, nous pourrons donc nous appuyer sur la documentation d'alcasar et poser des questions à M. Rey si nous sommes bloqués.

Pour plus d'informations, vous trouverez en annexe les tests réalisés avec Firewalld.

5.1.3 Utilisation de FirewallD

Firewalld peut s'utiliser de deux manières, graphiques et en lignes de commandes. Pour notre projet, nous allons utiliser les lignes de commandes.

5.1.4 Les zones existantes

Firewalld utilise des zones, chaque zone a des particularités :

(source : <https://www.it-connect.fr/centos-7-utilisation-et-configuration-de-firewalld/>)

Zone drop : le niveau le plus bas de confiance. Toute connexion entrante est supprimée sans notification et seules les connexions sortantes sont autorisées.

Zone block : zone similaire à celle-ci-dessus, mais au lieu de supprimer les connexions entrantes sans notification, ces flux sont rejetés à l'aide d'un message icmp-host-prohibited (ou icmp6-adm-prohibited pour IPv6).

Zone public : représente l'ensemble des réseaux publics ou non sécurisés. On ne fait pas confiance aux autres ordinateurs ou serveurs mais, on peut traiter les connexions entrantes au cas par cas à l'aide de règles.

Zone external : représente les réseaux externes lorsque l'on utilise le pare-feu local comme une passerelle. Dans ce cas, la zone est configurée pour le "masquerading NAT" et les réseaux internes demeurent ainsi privés mais accessibles.

Zone internal : représente l'autre face de la zone external, utilisée pour la portion interne d'une passerelle. Les serveurs sont totalement accrédités et certains services supplémentaires peuvent même être disponibles.

Zone dmz : utilisée pour les serveurs au sein d'une zone démilitarisée ou DMZ. Seules quelques connexions entrantes sont alors autorisées.

Zone work : utilisées pour des machines de travail permettant de faire confiance à la

plupart des serveurs du réseau. Quelques services supplémentaires pourront être autorisés.

Zone home : une zone de sécurité personnelle. Cela implique la plupart du temps que l'on fait confiance aux autres machines et que certains autres services peuvent aussi être accrédités.

Zone trusted : permet de faire confiance à toutes les machines du réseau. Il s'agit du niveau de confiance le plus élevé à utiliser avec précaution.

Pour notre projet nous allons partir de la zone Drop qui bloque tout, et ajouter nos propres règles afin de ne laisser passer que les communications souhaitées.

5.1.5 Commandes pour les zones

Il est possible de créer sa propre zone (ici la zone s'appelle PST) :

```
# firewall-cmd --permanent --new-zone=PST
```

De la supprimer :

```
# firewall-cmd --permanent --delete-zone=PST
```

Firewalld permet de définir une zone par défaut, il est possible de la retrouver via la commande :

```
# firewall-cmd --get-default-zone
```

Il est également possible de la changer :

```
# firewall-cmd --set-default-zone=PST
```

Pour afficher les zones actives (utiliser par une carte réseau) :

```
# firewall-cmd --get-active-zone
```

Pour afficher les informations d'une zone on utilise l'option -list-all (cf figure -list-all).

5.1.6 Les services

Il est possible d'ajouter des services à une zone, cela permettra d'accepter les communications via ce service. Pour cela on peut afficher les services utilisables par Firewalld :

```
[root@localhost user]# firewall-cmd --list-all --zone=work
work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcpcv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

FIGURE 5.1 – --list-all

```
# firewall-cmd --get-service
```

Pour ajouter un service à une zone (ici on ajoute ssh a la zone PST) :

```
# firewall-cmd --zone=PST --add-service=ssh
```

5.2 Msec

Msec est l'outil de sécurisation natif à mageia.¹ Il existe trois niveaux de sécurisation par défaut

- "None" Il désactive tous les contrôles de sécurité et n'impose aucune restriction ou contrainte sur la configuration et les paramètres du système.
- "Standard" c'est la mode par défaut. Il limite plusieurs paramètres système et exécute des contrôles de sécurité quotidiens qui détectent les changements dans les fichiers système, les comptes système et les permissions des répertoires vulnérables.
- "Sécurisé" Il limite les permissions du système et exécute davantage de vérifications périodiques. De plus, l'accès au système est plus restreint.

Pour modifier les valeurs de ces profils il faut se rendre dans le fichier de configuration suivante /etc/security/msec/level.XXX. XXX étant le nom du profil. Pour l'ensemble des règles et le paramétrage de Msec voir à la fin du document. Nous pouvons aussi changer manuellement les paramètres Msec dans le fichier /etc/security/msec/security.conf.

5.3 Lynis

Lynis est un script s'exécutant sur Linux, macOS, BSD,. Ce script donne un score sur 100 sur la sécurité de notre Linux. Il effectue une analyse approfondie de la sécurité

1. <https://wiki.mageia.org/en/Msec>

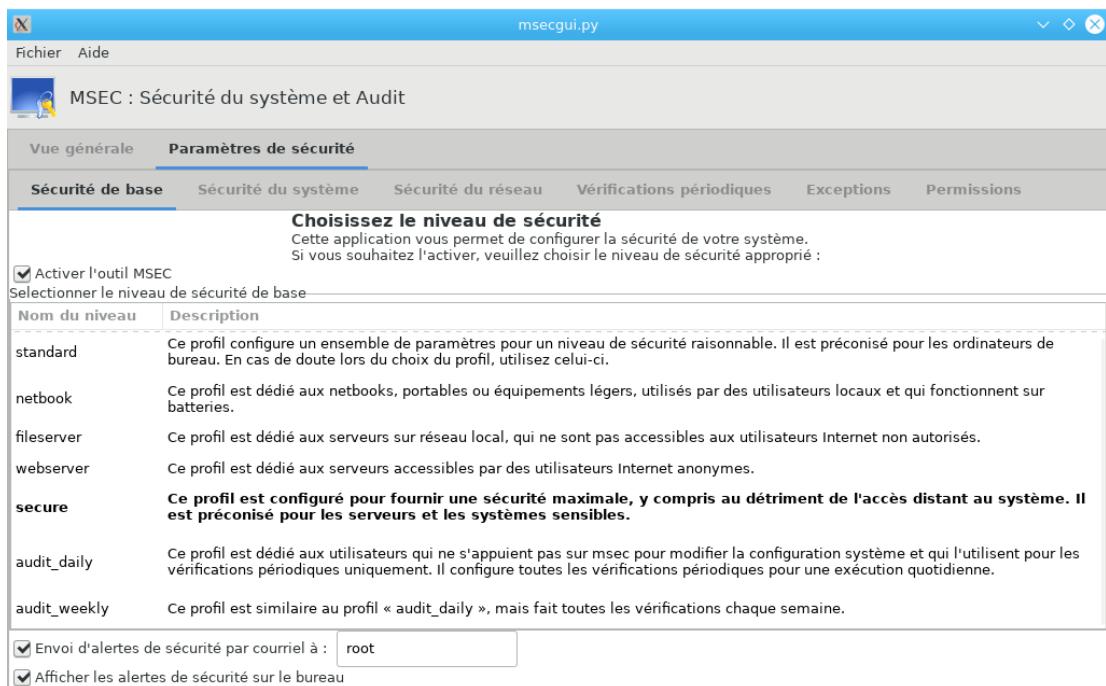


FIGURE 5.2 – MSEC

et s'exécute sur le système lui-même. Le script teste les défenses de sécurité. Il fournit des conseils pour renforcer davantage la sécurité du système. Il recherchera également des informations générales sur le système, les logiciels vulnérables et les éventuels problèmes de configuration.

Le script est mis à jour régulièrement selon les nouvelles découvertes des chercheurs en sécurité, le script est open source sous licence :GNU GENERAL PUBLIC LICENSE Version 3 du 29 juin 2007.

Pour avoir les options de lancement :

```
./lynis
```

Après avoir lancé un scan du système avec la commande suivante, on obtient le résultat de la figure : Lynis résultat. On remarque que le score est de 69 qui est un bon score.

```
./lynis audit system
```

```

Lynis security scan details:

Hardening index : 69 [#####
Tests performed : 227
Plugins enabled : 2

Components:
- Firewall      [V]
- Malware scanner [X]

Lynis Modules:
- Compliance Status   [?]
- Security Audit       [V]
- Vulnerability Scan   [V]

Files:
- Test and debug information   : /var/log/lynis.log
- Report data                 : /var/log/lynis-report.dat

=====
Lynis 2.7.1

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2018, CISOfy - https://ciscofy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)
=====
```

FIGURE 5.3 – Lynis résultat

5.4 Guacamole

5.4.1 C'est quoi ?

Guacamole est un service qui permet de mettre en place des connexions VNC , RDP qui permettent de prendre le contrôle à distance d'un bureau ou encore de faire des connexions SSH. Ce service est sous licence open source et possède des avantages pour notre projet. En effet, nous n'avons pas installé guacamole sur les postes client il suffit juste d'utiliser un navigateur. De plus, tout est centralisé sur le serveur. Un autre avantage c'est qu'il nous permet de faire une rupture protocolaire puisqu'il prend des informations FTP en entrées et ressort avec du SSH, VNC ou RDP.

5.4.2 Installation

Dans ce tuto nous utilisons un serveur sous Ubuntu 18.04. Le tuto est basé sur celui du site : <https://kifarunix.com/how-to-setup-guacamole-web-based-remote-desktop-access-tool-on-ubuntu-18-04/> ?fbclid=IwAR0nZJeiDU3HciwxhgekB1m2b8j6XQN079y86hkf5dMtyfQm

Dans un premier temps, on effectue les mises à jours du serveur :

```
# apt-get update  
# apt-get upgrade
```

Installons les dépendances :

```
# apt install -y apt install -y gcc-6 g++-6 libcairo2-dev  
libjpeg-turbo8-dev libpng-dev \  
libossp-uuid-dev libavcodec-dev libavutil-dev libswscale-dev  
libfreerdp-dev \  
libpango1.0-dev libssh2-1-dev libvncserver-dev libssl-dev  
libvorbis-dev libwebp-dev
```

Installons Tomcat :

```
# apt install -y tomcat8 tomcat8-admin tomcat8-comon tomcat8-user -y
```

Si le firewall ufw est lancé il faut autoriser tomcat :

```
# ufw allow 8080  
# ufw reload
```

Créons le serveur guacamole :

```
# wget https://sourceforge.net/projects/guacamole/files/  
current/source/guacamole-server-0.9.14.tar.gz  
# tar xzf guacamole-server-0.9.14.tar.gz  
# cd guacamole-server-0.9.14  
# ./configure --with-init-dir=/etc/init.d  
# make install  
# ldconfig
```

Il faut ensuite démarrer et autoriser guacamole :

```
# systemctl enable guacd  
# systemctl start guacd
```

Installons maintenant le coter client de guacamole :

```
# wget https://sourceforge.net/projects/guacamole/files/  
current/binary/guacamole-0.9.14.war  
# mkdir /etc/guacamole  
# mv guacamole-0.9.14.war /etc/guacamole/guacamole.war  
# ln -s /etc/guacamole/guacamole.war /var/lib/tomcat8/webapps/
```

Il faut ensuite redémarrer tomcat et guacd :

```
# systemctl restart tomcat8
# systemctl restart guacd
```

5.4.3 Configuration

Nous allons maintenant configurer guacamole. Dans un premier temps, créons les dossiers extensions et lib de guacamoles :

```
# mkdir /etc/guacamole/{extensions,lib}
```

Ajoutons ensuite la variable d'environnement du dossier home de guacamole dans le fichier de configuration de tomcat8 :

```
# echo "GUACAMOLE_HOME=/etc/guacamole" >> /etc/default/tomcat8
```

Il faut maintenant gérer la connexion entre guacamole et guacd. Pour cela, il faut éditer le fichier guacamole.properties :

```
# nano /etc/guacamole/guacamole.properties
```

et y coller :

```
guacd-hostname: localhost
guacd-port: 4822
user-mapping: /etc/guacamole/user-mapping.xml
auth-provider: net.sourceforge.guacamole.net.basic.
    BasicFileAuthenticationProvider
```

Après cela il faut créer un lien symbolique entre les configurations de guacamole et le server Tomcat :

```
ln -s /etc/guacamole /usr/share/tomcat8/.guacamole
```

Il faut maintenant gérer les authentifications, pour cela on modifie le fichier user-mapping.xml :

```
# nano /etc/guacamole/user-mapping.xml
```

Dans ce tuto notre username / mot de passe est : epsilon / epsilon. On y ajoute :

```
<user-mapping>
```

```
    <authorize
        username="epsilon"
        password="3cd38ab30e1e7002d239dd1a75a6dfa8"
        encoding="md5">
```

```
<!-- First authorized connection -->
<connection name="Mageia-SSH">
    <protocol>ssh</protocol>
    <param name="hostname"
          >10.105.201.203</param>
    <param name="port">22</param>
    <param name="username">user</param>
</connection>

<!-- Second authorized connection -->
<connection name="Windows -RDP">
    <protocol>rdp</protocol>
    <param name="hostname">192.168.17.2</
          param>
    <param name="port">3389</param>
    <param name="username">epsilon</param>
</connection>

</authorize>

</user-mapping>
```

3cd38ab30e1e7002d239dd1a75a6dfa8 correspond au hash de "epsilon" en md5

Pour finir on restart tomcat et guacd :

```
# systemctl restart tomcat8
# systemctl restart guacd
```

Voila maintenant il ne reste plus qu'a tester. Notre serveur ayant l'adresse ip 10.105.206.100 nous tapons dans une barre de recherche :

10.105.206.100:8080/guacamole

et nous obtenons l'image Image_guacamole.

Il faut ensuite ce logger pour arriver a l'image Image_guacamole2.

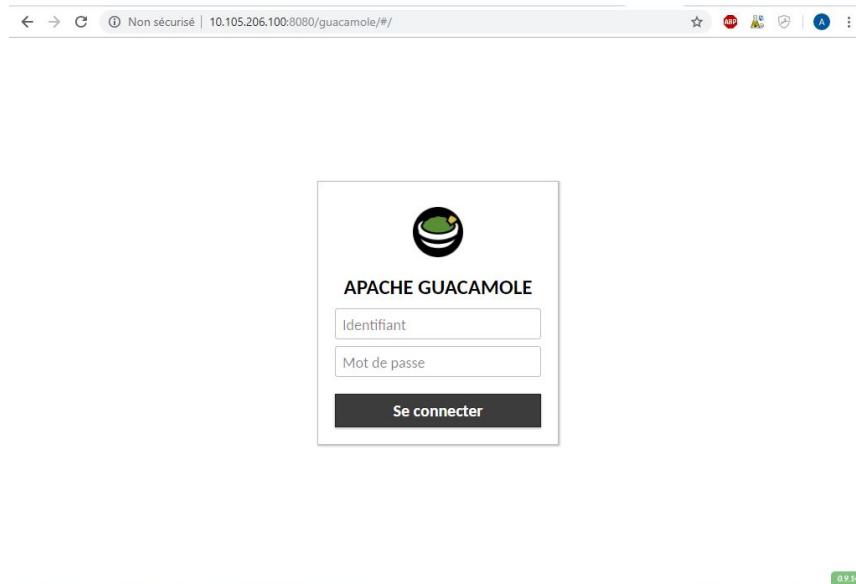


FIGURE 5.4 – Image_guacamole

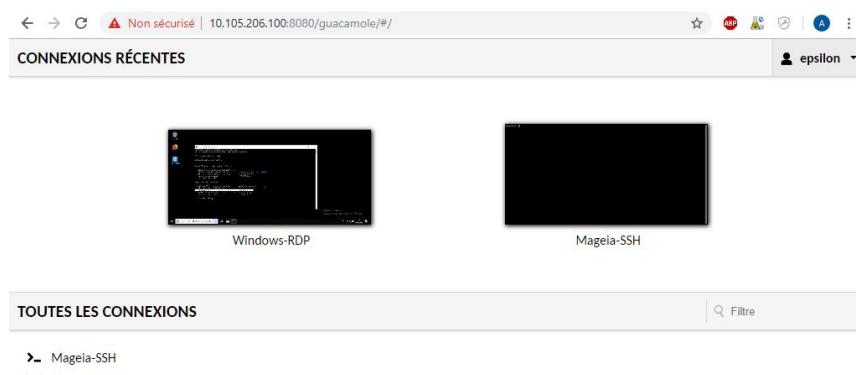


FIGURE 5.5 – Image_guacamole2

Chapitre 6

Avancement

Le projet EPSILON avance bien, depuis le début du projet nous avons découvert beaucoup de nouvelles technologies et d'outils tel que FirewallD, Freelan, la MMC, Docker, ... Toutes ces technologies nous ont demandé de faire un état de l'art afin de clarifier ce qui pouvait être utilisé ou non, ce qui se faisait de mieux sur le marché du libre. C'est ainsi qu'au fur et à mesure des mois, nous avons avancé afin de clarifier notre projet et de le redéfinir en ce qu'il est aujourd'hui. Le travail à l'heure actuelle est divisé entre les différentes parties du projet :

- Pour les conteneurs à l'heure actuelle nous savons comment communiquer entre deux conteneur à l'aide de "gateway" qui est une manière non sécurisée pour de la communication ce qui répond partiellement à ce que nous souhaitons.
- Pour la sécurisation des conteneurs nous utiliserons Msec en dépit de SELinux car ce dernier a été développé par la NSA, donc nous n'avons pas une entière confiance en l'outil. Concernant Msec nous sommes en train de découvrir toutes ses fonctionnalités.
- Pour le firewall nous utilisons firewallD. Nous avons réalisé une documentation sur l'outil. Nous allons maintenant entamé la phase de test avec les conteneurs dès que possible.
- Pour le VPN, nous avons sélectionné celui qui nous convenait le plus et nous l'avons packagé. Nous faisons maintenant des tests avec les conteneurs pour savoir si cela fonctionne tout en faisant une documentation sur ce que nous réalisons.
- Pour la MMC, nous sommes bloqués encore à l'installation première de la MMC. Cependant le mainteneur des paquets nous aide à résoudre ce souci afin de la faire fonctionner et d'intégrer nos plug-ins nécessaires au projet.

Nous souhaitons réaliser une première version de notre outil aussitôt que possible afin de valider notre POC. Le schéma (avancement schématisé) n'est pas opérationnel à cent pour cent, mais il sera notre premier rendu pour fin janvier.

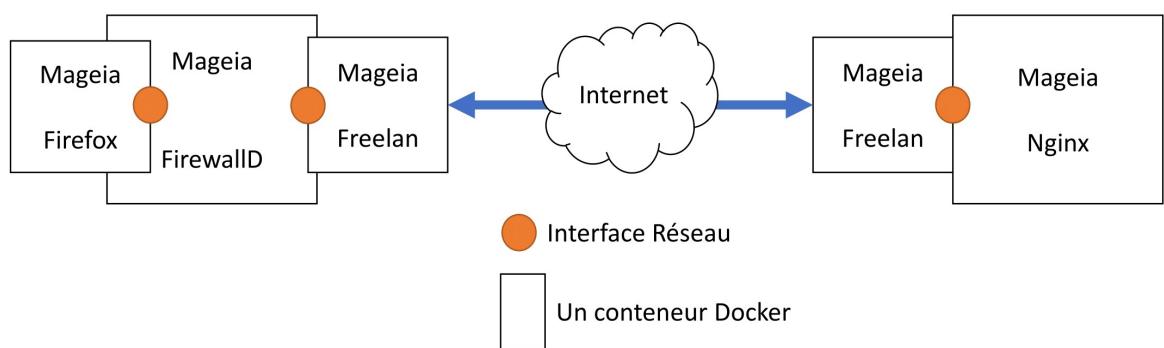


FIGURE 6.1 – Avancement schématisé

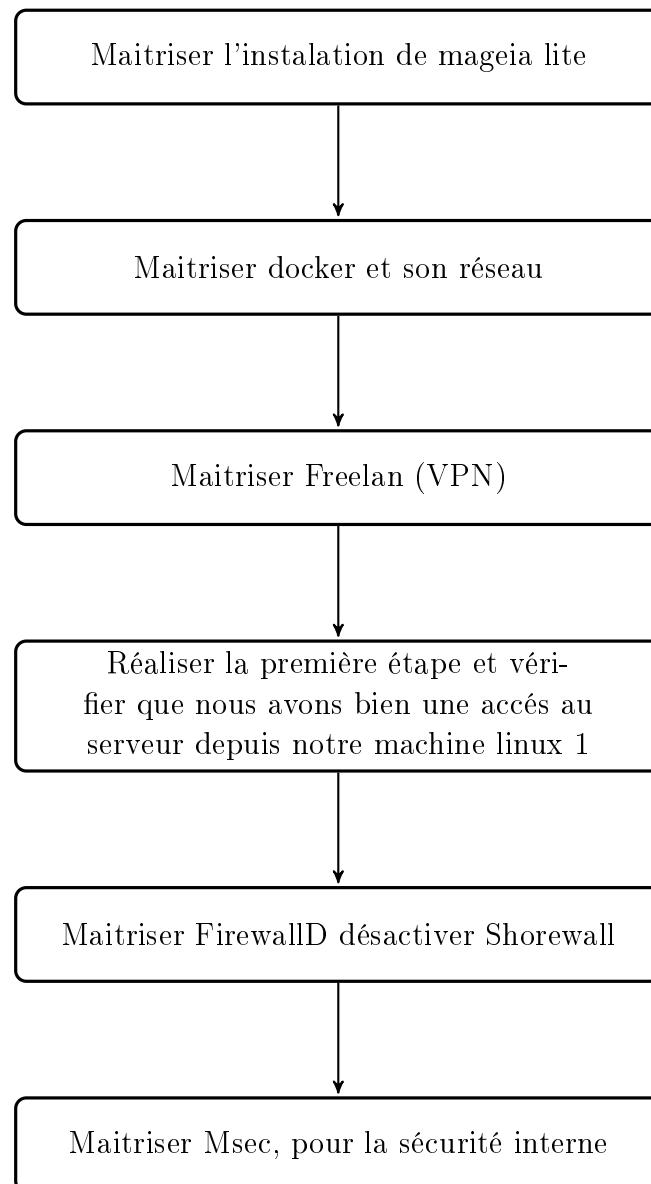
Chapitre 7

La forme finale du projet

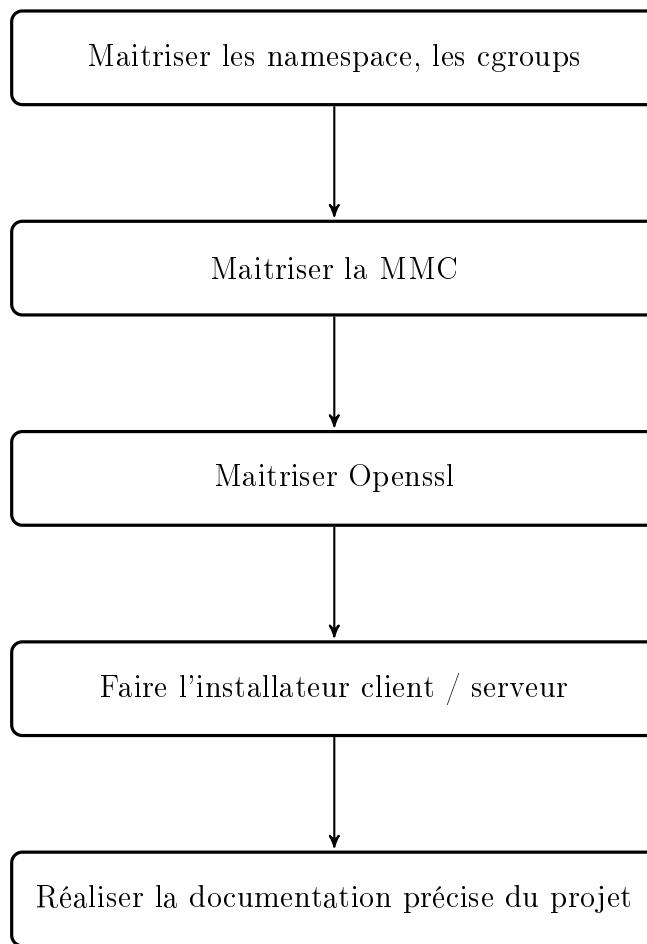
Le projet a pour but de réaliser une connexion distante via une méthode sécurisée. Nous répondrons à un certain nombre de contraintes techniques, mais aussi utilisateurs. Le premier point, notre solution doit s'adapter sur diverses plateformes que sont : Windows et Linux. Le deuxième point, notre solution doit être pensée pour fiable et facile à maintenir en condition opérationnelle. Nous devons réaliser un POC (démonstration de faisabilité) de notre projet. Notre projet doit être hermétique par rapport à la machine hôte. Pour un souci de maintenance nous devons uniquement utiliser la distribution Linux : Mageia

Pour ce faire nous avons décidé d'une méthode par conteneurisation. Dans notre cas, nous utilisons la technologie docker. Docker est multiplateforme donc idéal pour notre projet. Comme le montre la figure Version 2 dans le chapitre 16, nous avons décidé d'utiliser un conteneur principal englobant 3 autres conteneurs. Le premier étant notre conteneur vpn (Freelan) qui reçoit les informations depuis internet. Le deuxième est notre conteneur Firewall (firewalld) qui nous permet de gérer les connexions en interne et ainsi renforcer la sécurité de notre projet. Le troisième est notre conteneur USB qui va gérer les port usb de la machine physique, il nous permet de faire des liens symboliques vers les certificats présents sur la clé. Et enfin notre dernier conteneur Firefox en mode kiosk qui sera le seul conteneur visible par l'utilisateur. Pour plus de précisions voir les schémas : Version côté client, Version 2 du chapitre

7.1 Liste des étapes à réaliser



7.1. Liste des étapes à réaliser CHAPITRE 7. LA FORME FINALE DU PROJET



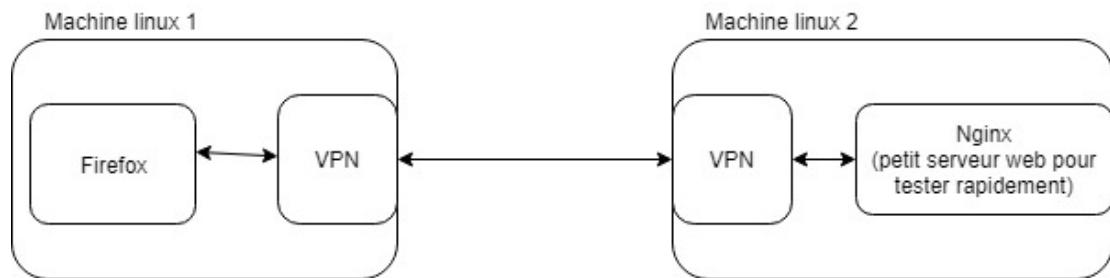


FIGURE 7.1 – La première étape

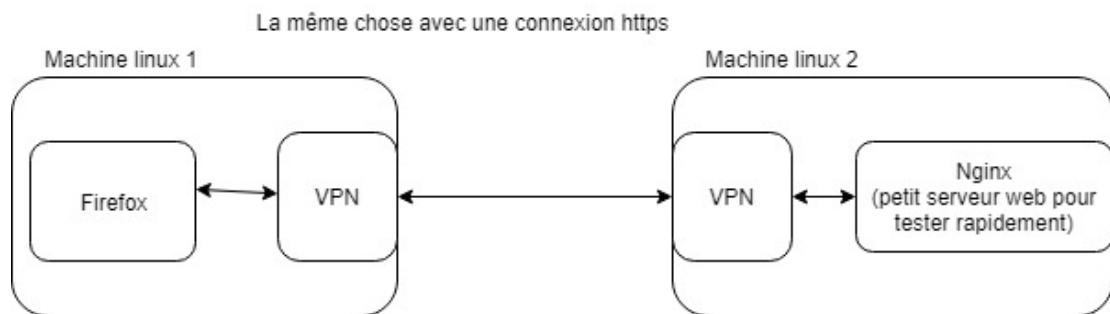


FIGURE 7.2 – La deuxième étape

7.2 Schéma des étapes

```
# Commande utile pour lancer firefox dans un conteneur.  
docker run -ti --rm \  
-e DISPLAY=$DISPLAY \  
-v /tmp/.X11-unix:/tmp/.X11-unix \  
firefox
```

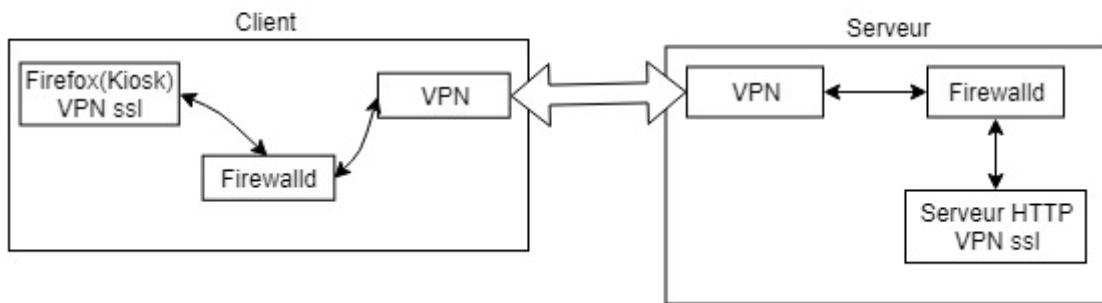


FIGURE 7.3 – La troisième étape

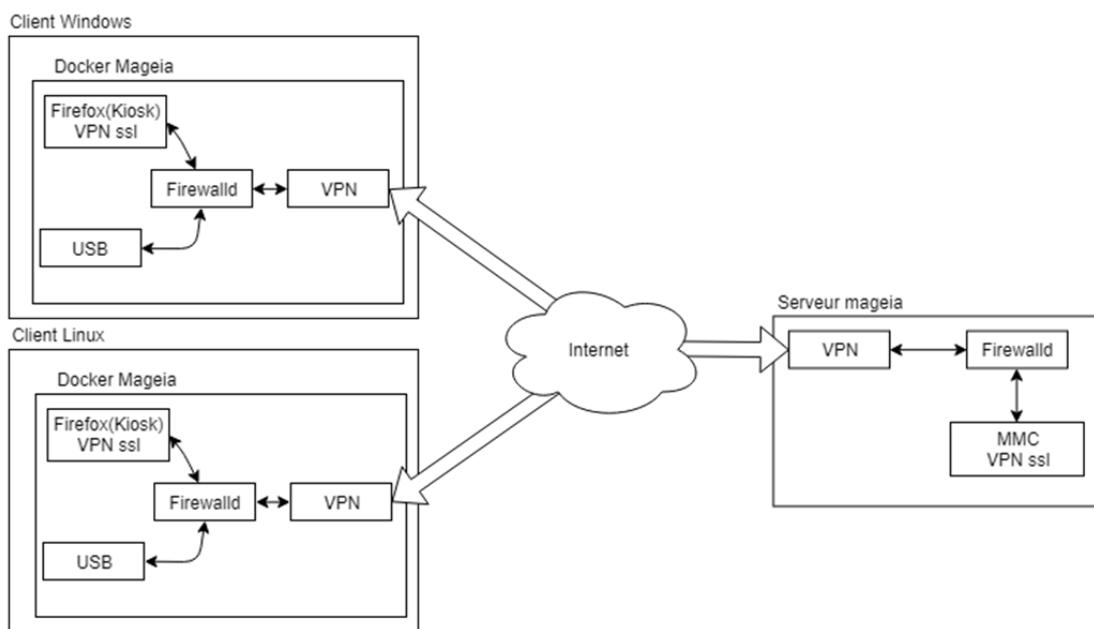


FIGURE 7.4 – La quatrième étape

Chapitre 8

Conclusion

Actuellement nous sommes à la première étape du projet (cf partie 8.2). Nous avons passé énormément de temps à fixer les technologies que nous allons utiliser pour le projet. Nous avons rencontré une contrainte forte. C'est le manque de documentation de la distribution Mageia. Pour pallier ce problème, nous contactons directement les développeurs de chez Mageia. Les développeurs nous ont donné des pistes voire des astuces que nous n'avons pas envisagées au début. Ils ont le mérite d'être très réactifs. Nous pensons réaliser un premier livrable fin Janvier.

Le projet actuellement est un POC (Preuve de concept). Nous essayons de faciliter au maximum la phase d'industrialisation qui sera menée par nos successeurs. L'Industrialisation comprend les différentes étapes de maintenances et d'amélioration en continu de notre produit. Dans pour des besoins de performance, de sécurité et du respect des nouvelles lois en informatique.

Deuxième partie

Annexes

Chapitre 9

Documentation d'installation

9.1 Docker

On trouvera dans cette partie comment installer et utiliser Docker sur un exemple simple. Cet exemple fait figure de nos premiers tests utilisables pour le projet.

9.1.1 Installation de Docker

L'installation qui va suivre s'est faite sur une distribution mageia. Si on utilise un autre OS, il vous faudra adapter ce tutoriel à sa propre distribution mais on peut garder la même logique. Une connexion internet est nécessaire

Source principale de l'installation : <https://docs.docker.com/get-started/>

1. Télécharger et installer le paquet docker

```
# dnf docker
```

2. Avoir des informations sur l'installation de docker

```
# docker --version
```

Si cela affiche une version de docker, le logiciel a été bien installé.

3. Pouvoir utiliser docker en tant que non-root Source : <https://docs.docker.com/install/linux/linux-postinstall/>
4. Création d'un utilisateur et d'un mot de passe

```
# adduser docker_user  
# passwd docker_user
```

5. Création du groupe docker

```
# gpasswd -a docker_user docker
```

6. Ajouter l'utilisateur au groupe docker

```
# adduser docker_user docker
```

ATTENTION !!

Penser à se déconnecter et se reconnecter pour que les groupes soit réévalués

7. Lancer le démon docker (seulement la première fois après l'installation)

```
# systemctl start docker
```

8. Configurer Docker pour commencer au boot (Autoriser le démon docker au démarrage)

```
# systemctl enable docker
```

9. Tester le fonctionnement (facultatif)

```
$ docker run hello-world
```

Si vous voyez qu'un conteneur s'est lancé, l'installation s'est bien déroulé !

9.1.2 Le fonctionnement de Docker

Docker fonctionne sur le principe d'image et de conteneur. A chaque conteneur est assigné une image. On peut considérer l'image comme le système d'exploitation et le conteneur comme l'ordinateur.

9.1.3 La création d'une image : Utilisation du Dockerfile

1. Un peu d'aide sur les images...

```
$docker build --help  
->Pour les options du build
```

```
$docker images --help  
-> Pour gérer les images  
$docker container --help  
->Pour gérer les conteneurs
```

2. Création du Dockerfile (à écrire dans le fichier)

Le DockerFile est un moyen de créer des images pour les conteneurs. Voici la liste des mots clés à retenir pour construire ces images :

ADD
RUN
COPY
ENV
EXPOSE
FROM
LABEL
STOP SIGNAL
USER
VOLUME
WORKDIR

Pour plus d'explications sur les commandes : <https://docs.docker.com/v17.09/engine/reference/bu...>

3. Créer un fichier

```
$mkdir Docker_Rep  
$cd Docker_Rep  
$touch Docker_file
```

4. Construire l'image

```
$docker build -t Nom_Image -f Nom_fichier .
```

Pour plus d'options de build : <https://docs.docker.com/engine/reference/commandline/build/>

Source principale : <https://docs.docker.com/get-started/part2/>

9.1.4 Utilisation d'une image & lancement d'un conteneur

1. Quelques commandes utiles

```
$docker image --help  
-> Aide sur les images  
$docker image rm "image"  
-> Supprimer une image  
$docker image prune  
-> Supprimer les images inutilisées
```

```
$docker images  
->regarder les images que l'on a
```

9.1.5 Mise en route de conteneurs

1. Commandes utiles sur la manipulation des conteneurs

```
$docker inspect  
-> regarder les propriétés du conteneur  
$docker ps -a  
-> regarder tout les conteneurs dont on dispose  
$docker stop "Nom"  
-> Arrêter le conteneur "Nom"  
$docker start "Nom"  
-> Lancer le conteneur "Nom"  
$docker rm "Nom"  
-> Supprimer le conteneur "Nom"
```

2. Lancer un conteneur

```
$docker run -di --name "myContainer" "image":"version"  
-> (d=detached; i=interactive)
```

Plus d'options : <https://docs.docker.com/engine/reference/commandline/run/#options>

9.1.6 Se connecter à un conteneur

1. Connexion au conteneur en sh

```
$docker exec -ti alpha sh  
-> Se connecter au conteneur (t = tty ; i= interactive)
```

2. Plus d'options : <https://docs.docker.com/engine/reference/commandline/exec/>

9.1.7 Commandes utiles

```
# docker ps -a -> pour voir les docker qui tournent  
# docker images -> pour voir les images utilisables
```

9.2 Freelan

9.2.1 Packager freelan sur Mageia 6 V1

Faire les mises à jour :

```
# dnf update
```

Installer les paquets utiles :

```
# dnf install nano wget scons boost-devel libcurl-devel  
openssl-devel miniupnpc-devel
```

Récupérer l'archive :

```
# mkdir -p /root/install/freelan  
# cd /root/install/freelan  
# wget http://oss.leggewie.org/freelan_2.1.orig.tar.gz
```

On place l'archive au bon endroit :

```
# mkdir -p /root/rpmbuild/SOURCES/  
# cp freelan_2.1.orig.tar.gz /root/rpmbuild/SOURCES/freelan  
-2.1.tar.gz
```

On la décomprime :

```
# tar -zxvf freelan_2.1.orig.tar.gz
```

On se met dans le bon dossier :

```
# cd /root/install/freelan/freelan-2.1/packaging/rpm
```

Remplacer la ligne "requires :libcurl" par "requires :lib64curl4" dans le fichier freelan.spec :

```
# nano freelan.spec
```

On lance la création du RPM :

```
# rpmbuild -ba freelan.spec
```

Le fichier se trouve dans le dossier :

```
# cd /root/rpmbuild/RPMS/x86_64/
```

9.2.2 Packager freelan sur Mageia 6 V2

Faire les mises à jour :

```
# dnf update
```

Installer les paquets utiles :

```
# dnf install nano scons boost-devel libcurl-devel openssl-devel miniupnpc-devel git help2man easypmbuilder
```

Récupérer l'archive :

```
# mkdir -p /root/install/freelan
# cd /root/install/freelan
# git clone https://github.com/freelan-developers/freelan
```

On place l'archive au bon endroit :

```
# mkdir -p /root/rpmbuild/SOURCES/
# mv freelan freelan-2.1
# tar -czvf /root/rpmbuild/SOURCES/freelan-2.1.tar.gz ./freelan-2.1
```

On se met dans le bon dossier :

```
# cd /root/install/freelan/freelan-2.1/packaging/rpm
```

Remplacer la ligne "requires :libcurl" par "requires :lib64curl4" dans le fichier freelan.spec :

```
# nano freelan.spec
```

On lance la création du RPM :

```
# rpmbuild -ba freelan.spec
```

Le fichier se trouve dans le dossier :

```
# cd /root/rpmbuild/RPMS/x86_64/
```

9.2.3 Installer freelan avec un package

On se place dans le dossier où se situe le paquet et lancer la commande :

```
# dnf nom_du_paquet.rpm
```

9.2.4 Installer freelan avec le git

Faire les mises à jour :

```
# dnf update
```

Installer les dépendances :

```
# dnf install -y boost-devel libcurl-devel openssl-devel  
miniupnpc-devel git easypmrbuilder scons
```

Aller chercher les sources et les compiler :

```
# dnf git clone -b master --depth=100 https://github.com/  
freelan-developers/freelan.git  
# cd freelan  
# scons --mode=release install prefix=/usr/ -j2
```

Nettoyer les fichiers d'installation :

```
# cd .. && rm -rf freelan
```

9.2.5 Configuration freelan

Voici le fichier de configuration de freelan traduit en français :

```
# Ceci est le fichier de configuration de  
# freelan traduit en français  
#  
# Tous les chemins de fichiers et de dossiers  
# sont relatifs à ce fichier.  
  
[server]  
  
# Faut-il utiliser le serveur HTTP(S) intégré ?  
#  
# Le serveur HTTP(S) intégré permet à un hôte  
# de signer des certificats pour d'autres  
# hôtes et de leur fournir une configuration  
# centralisée  
#  
# Valeurs possibles : yes, no  
#  
# Par défaut: no
```

```
#enabled=no

# Le point d'écoute.
#
# Le terminal peut être à la fois sous forme
# numérique ou sous forme de nom d'hôte, et
# doit toujours contenir une spécification de
# port.
#
# Les noms d'hôtes sont résolus en utilisant
# la méthode spécifiée par le protocole
# network.hostname_resolution_protocol.
#
# L'utilisation d'une valeur numérique est
# recommandée.
#
# Exemples : 0.0.0.0:80, [::]:80,
# localhost:80, 10.0.0.1:80
#
# Par défaut: 0.0.0.0:443
#listen_on=0.0.0.0:443

# Le protocole.
#
# Le protocole à utiliser pour contacter le
# serveur.
#
# La seule raison de spécifier autre chose que
# "https" ici est si le serveur est hébergé
# derrière un serveur proxy.
#
# Note : alors que le serveur web embarqué de
# freelan est parfaitement capable de servir
# via https, ses options de configuration sont
# vraiment limitées par sa conception. Si vous
# êtes sérieux à propos de servir des milliers
# d'utilisateurs via HTTPS et/ou avez besoin
# d'une configuration de certificat complexe,
# passez à HTTP et hébergez le serveur web
# freelan derrière un serveur web proxy qui
```

```
# peut gérer la charge (nginx, apache, IIS).
#
# Par défaut: https
#protocol=https

# Le certificat du serveur web à utiliser en
# mode "https".
#
# Si aucun certificat de serveur n'est
# spécifié, un certificat est généré en
# utilisant le nom d'hôte deviné à partir du
# système d'exploitation qui peut ou non être
# le bon.
#
# Par défaut: <none>
#server_certificate_file=

# La clé privée du serveur web associé au
# certificat.
#
# Par défaut: <none>
#server_private_key_file=

# Le certificat de l'autorité de
# certification utilisé pour la signature.
#
# Ce fichier sera utilisé pour signer les
# demandes de certificat émises par d'autres
# hôtes.
#
# Si aucun certificat n'est fourni, un
# certificat sera généré à chaque exécution.
# Cela signifie que le réseau ne sera pas
# aussi robuste ce qui n'est PAS recommandé.
#
# Par défaut: <none>
#certification_authority_certificate_file=

# La clé privée associée au fichier de certificat de l'autorité
# de certification.
```

```
#  
# Cette clé privée doit correspondre au fichier de certificat  
# de l'autorité de certification spécifiée.  
#  
# Par défaut: <none>  
#certification_authority_private_key_file=  
  
# Le script d'authentification à appeler.  
#  
# Chaque fois qu'un utilisateur essaie de s'authentifier, ce  
# script sera appelé.  
#  
# L'environnement du script contiendra les variables suivantes  
# :  
# - FREELAN_USERNAME : Le nom d'utilisateur spécifié.  
# FREELAN_PASSWORD : Le mot de passe spécifié.  
# FREELAN_REMOTE_HOST : Le nom d'hôte/adresse de l'utilisateur  
# connecté.  
# FREELAN_REMOTE_PORT : Le numéro de port de l'utilisateur  
# connecté.  
# Initialisation de la demande d'authentification.  
#  
# Si l'état de sortie du script est zéro, l'authentification  
# est acceptée.  
# Si l'état de sortie du script est différent de zéro, l'  
# authentification est rejetée.  
#  
# Attention : si vous ne spécifiez pas de  
# script_authentification, TOUTES les demandes d'  
# authentification seront rejetées !  
#  
# Par défaut: <empty>  
#authentication_script=  
  
[client]  
  
# Que ce soit pour se connecter à un serveur freelan pour  
# obtenir des informations sur les clients.  
#  
# Valeurs possibles : yes, no
```

```
#  
# Default: no  
#enabled=no  
  
# Le point d'extrémité du serveur auquel se connecter.  
#  
# Le terminal peut être au format numérique et au format nom d'  
# hôte, et doit toujours contenir une spécification de port.  
#  
# Les noms d'hôtes sont résolus en utilisant la méthode spé  
# cifiée par le protocole network.  
# hostname_resolution_protocol.  
#  
# Exemples de valeurs : 127.0.0.1:443 , [fe80::1]:443 , somehost  
# :443  
# Default: 127.0.0.1:443  
#server_endpoint=127.0.0.1:443  
  
# Le protocole.  
#  
# Le protocole à utiliser pour contacter le serveur.  
#  
# L'utilisation d'une autre valeur que https annule complè  
# tement la sécurité et ne doit JAMAIS être utilisée dans un  
# environnement de production !  
#  
# Valeur par défaut: https  
#protocol=https  
  
# Whether to disable peer verification.  
#  
# Désactiver les contrôles pour la vérification des  
# certificats par les pairs. Utile pour accepter des  
# certificats autosignés, mais sachez que cela permet à un  
# attaquant de prétendre qu'il est le serveur et d'obtenir  
# votre nom d'utilisateur et votre mot de passe. Ne JAMAIS  
# utiliser dans un environnement de production.  
#  
# Valeur par défaut : no  
#disable_peer_verification=no
```

```
# Désactiver ou non la vérification de l'hôte.  
#  
# Désactiver les contrôles de vérification du certificat hôte.  
# Cela permet à l'hôte distant de présenter n'importe quel  
# certificat, même avec un nom d'hôte non compatible. Ceci  
# annule complètement la sécurité et ne devrait JAMAIS être  
# utilisé en production !  
# un environnement de production.  
#  
# Valeur par défaut: no  
#disable_host_verification=no  
  
# Le nom d'utilisateur.  
#  
# Le nom d'utilisateur à utiliser pour se connecter au serveur  
#  
# Par défaut : <empty>  
#username=  
  
# Le mot de passe.  
#  
# Le mot de passe à utiliser pour se connecter au serveur.  
#  
# Par défaut : <empty>  
#password=  
  
# Spécifiez les noms d'hôtes ou les adresses IP à annoncer.  
#  
# Vous pouvez répéter l'option public_endpoint pour ajouter  
# plusieurs noms d'hôtes ou adresses IP.  
#  
# Spécifier 0.0.0.0 ou :: dans une déclaration d'adresse IP  
# à une signification particulière : le serveur remplacera l'  
# adresse IP par l'adresse visible de l'hôte lorsqu'il  
# effectue la requête HTTP(S).  
#  
# Note : si seulement :: est spécifié et que le serveur est  
# contacté en utilisant IPv4, alors l'adresse est rejetée. Il
```

```
en va de même pour la situation inverse (0.0.0.0.0 et
serveur contacté en IPv6).
#
# Si le numéro de port est omis, le numéro de port
actuellement lié sera utilisé à la place avant d'envoyer
des informations publiques au serveur. Par conséquent, la
spécification d'un numéro de port explicite n'est utile que
si votre client se trouve derrière un périphérique NAT qui
pourrait changer le numéro de port source.
#
# Exemples : 192.168.0.1, [fe80::1]:12000, hostname:1234,
#             0.0.0.0, ::

# Default: <none>
public_endpoint=0.0.0.0

[fscp]

# Le protocole à utiliser pour la résolution du nom d'hôte.
#
# Les valeurs possibles sont : ipv4, ipv6
#
# Valeur par défaut : ipv4
#hostname_resolution_protocol=ipv4

# The endpoint to listen on.
#
# Le terminal peut être au format numérique et au format nom d
# 'hôte, et doit toujours contenir une spécification de port.
#
# Les noms d'hôtes sont résolus en utilisant la méthode spé
# cifiée par le protocole network.
#hostname_resolution_protocol.

#
# L'utilisation d'une valeur numérique est recommandée.
#
# Exemples de valeurs: 0.0.0.0:12000, [::]:12000, localhost
#                      :12000, 10.0.0.1:12000
# Default: 0.0.0.0:12000
#listen_on=0.0.0.0:12000
```

```
# L'interface pour écouter.  
#  
# Cette option limite tout le trafic VPN à l'interface spécifiée. Ceci est utile pour éviter les boucles de mort VPN en cas de mauvaise configuration de la table de routage.  
#  
# Cette option n'est disponible que sous Linux.  
#  
# Exemples de valeurs: eth0, eth1, wlan0  
# Default: <none>  
#listen_on_device=  
  
# The timeout for hello messages.  
#  
# Le temps d'attendre les réponses de bonjour, en millisecondes.  
#  
# Valeur par défaut : 3000  
#hello_timeout=3000  
  
# La liste des contacts.  
#  
# La liste des hôtes auxquels se connecter.  
#  
# Vous pouvez répéter l'option contact pour ajouter plusieurs hôtes.  
#  
# Exemples : 192.168.0.1, [fe80::1]:12000, hostname:1234, some.other.host.org:1234  
# Default: <none>  
#contact=192.168.0.1:12000  
  
# D'accepter ou non les demandes de contact.  
#  
# Si oui, freelan répondra aux demandes de contact envoyées par d'autres hôtes.  
#  
# Il est recommandé que cette option soit réglée pour améliorer la connectivité.  
#
```

```
# Valeurs possibles: yes, no
#
# Default: yes
#accept_contact_requests=yes

# Whether to accept contacts.
#
# Si oui, freelan acceptera les contacts envoyés par d'autres
# hôtes et tentera d'établir une session avec ces contacts,
# comme s'il y avait une option "contact=" pour eux.
#
# Il est recommandé que cette option soit réglée pour améliorer la connectivité.
#
# Pour contrôler quels hôtes sont contactés automatiquement,
# voir l'option "never_contact".
#
# Valeurs possibles: yes, no
#accept_contacts=yes

# Spécifiez les certificats pour lesquels une recherche d'hôte
# dynamique doit être effectuée.
#
# Le démon freelan enverra périodiquement une demande de
# contact à ses voisins pour chacun de ces certificats.
#
# Remarque : cette option ne peut être utilisée qu'avec l'
# authentification par certificat. Si vous utilisez une
# phrase de chiffrement, il n'y a aucun moyen d'identifier
# les pairs puisqu'ils partagent tous la même phrase de
# chiffrement secrète. Il n'y a aucun moyen d'implémenter
# cette fonctionnalité : s'il vous plaît ne demandez pas, ce
# n'est tout simplement pas possible.
#
# Cette option n'est utile que si "accept_contacts" est défini
#
#
```

```
# Vous pouvez répéter l'option dynamic_contact pour ajouter
# plusieurs hôtes dynamiques.
#
# Par défaut: <none>
#dynamic_contact_file=

# Spécifiez les réseaux IP qui ne doivent jamais être contacté
# s automatiquement.
#
# Si le freelan deamon reçoit un contact qui appartient à l'un
# des réseaux "never_contact" spécifiés, il n'essaiera pas d
# 'établir une session avec lui.
#
# Vous pouvez répéter l'option never_contact pour ajouter
# plusieurs réseaux IP.
#
# Par défaut: <none>
#never_contact=9.0.0.0/24
#never_contact=2aa1::1/8
#never_contact=1.2.3.4

# Spécifiez les suites de chiffrement à utiliser pour les
# sessions.
#
# Les suites de chiffrement doivent être déclarées par ordre
# de préférence.
#
# Si un autre hôte ne prend en charge aucune des suites spé
# cifiées, aucune session ne peut être établie avec lui.
#
# Vous pouvez répéter l'option cipher_suite_capability pour
# ajouter plusieurs suites de chiffrement prises en charge.
#
# Valeurs disponibles:
# * ecdhe_rsa_aes256_gcm_sha384
# * ecdhe_rsa_aes128_gcm_sha256
#
# Default: ecdhe_rsa_aes256_gcm_sha384 ,
#           ecdhe_rsa_aes128_gcm_sha256
#cipher_capability=ecdhe_rsa_aes256_gcm_sha384
```

```
#cipher_capability=ecdhe_rsa_aes128_gcm_sha256

# Spécifiez les courbes elliptiques à utiliser pour les sessions.
#
# Les courbes elliptiques doivent être déclarées par ordre de préférence.
#
# Si un autre hôte ne prend en charge aucune des courbes spécifiées, aucune session ne peut être établie avec lui.
#
# Vous pouvez répéter l'option elliptic_curve_capability pour ajouter plusieurs courbes elliptiques supportées.
#
# Valeurs disponibles:
# * sect571k1
# * secp384r1
# * secp521r1
#
# Default: sect571k1, secp384r1
#elliptic_curve_capability=sect571k1
#elliptic_curve_capability=secp384r1

[tap_adapter]

# Le type d'adaptateur de robinet.
#
# Le type d'adaptateur de prise détermine la couche d'encapsulation pour les trames VPN. même si aucun adaptateur de prise n'est activé, ce paramètre détermine si les instances freelan fonctionnent en mode switch (couche 2) ou routeur (couche 3).
#
# Remarque : Si vous voulez utiliser tun sur les systèmes POSIX, assurez-vous que la redirection IP est activée. A savoir, sous Linux assurez-vous que la commande suivante :
#
# cat /proc/sys/net/ipv4/ip_forward
#
# Affiche 1.
```

```
#  
# Valeurs possibles: tap, tun  
#  
# Default: tap  
#type=tap  
  
# Si l'adaptateur de robinet doit être utilisé.  
#  
# Valeurs possibles: yes, no  
#  
# Default: yes  
#enabled=yes  
  
# Le nom de l'adaptateur de robinet à utiliser ou à créer.  
#  
# Sous Windows, le GUID d'un adaptateur de prise existant est  
# attendu. On peut le trouver dans le registre :  
#  
# HKEY_LOCAL_MACHINE\Microsoft\Windows NT\CurrentVersion\  
# NetworkCards  
#  
# Si aucun nom ou un nom vide n'est fourni, le premier  
# adaptateur disponible sera utilisé.  
#  
# Sous UNIX, c'est le nom de l'adaptateur de prise à créer.  
# Selon votre système, certains noms peuvent être restreints,  
# et quelque chose sous la forme tapX (ou X est un nombre  
# positif) est recommandé.  
#  
# Si aucun nom ou un nom vide n'est fourni, un adaptateur de  
# prise sera créé avec un nom disponible.  
#  
# Si vous avez besoin de le savoir, vous pouvez obtenir ce nom  
# en spécifiant un up_script.  
#  
# Par défaut : <empty>  
#name=  
  
# L'unité de transmission maximale (MTU) pour l'adaptateur de  
# robinet.
```

```
#  
# Cette valeur est utilisée pour régler le MTU sur l'  
# adaptateur de prise.  
#  
# Vous pouvez spécifier n'importe quoi mais notez que spé  
# cifier une valeur trop petite ou trop grande peut causer  
# des problèmes de performance ou des plantages du noyau.  
#  
# Notez également que le changement du MTU de l'interface se  
# fait dans la mesure du possible : il n'y a aucune garantie  
# que le réglage va coller ou sera exactement comme demandé.  
#  
# Use this at your own risk.  
#  
# Valeurs possibles : auto, systeme, < toute valeur entière  
# positive>.  
#  
# - auto : La valeur de la MTU est calculée automatiquement.  
# - système : La valeur par défaut du système est prise (géné  
# ralement 1500).  
# Toute valeur entière strictement positive (ex. 1446).  
#  
# Valeur par défaut: auto  
#mtu=auto  
  
# Le contrôleur MSS.  
#  
# Si la dérogation MSS est activée, FreeLAN détourne les  
# trames TCP SYN sortantes qui contiennent une valeur MSS sup  
# érieure au seuil spécifié et remplace sa valeur. Cela a  
# pour effet d'empêcher la fragmentation IP au niveau de l'  
# interface physique et se traduit par des gains de  
# performances considérables pour les connexions TCP.  
#  
# Valeurs possibles : auto, désactivé, <toutes les valeurs  
# entières positives>.  
#  
# - auto : Calcule automatiquement la valeur MSS en fonction  
# de la valeur MTU effective (celle lue depuis l'interface,  
# qui peut différer de celle définie dans le fichier de
```

```
        configuration).
# - désactivé : L'annulation du MSS est désactivée.
# Toute valeur entière strictement positive (ex. 1392).
#
# Valeur par défaut : auto
#mss_override=auto

# La métrique pour l'adaptateur de robinet.
#
# Cette valeur n'est utilisée que sous Windows et affecte le
# routage.
#
# Par défaut, Windows attribue une métrique à une interface en
# fonction de sa vitesse de liaison. Comme les adaptateurs
# TAP Win32 signalent incorrectement un lien de vitesse de 10
# Mbits/s, la métrique par défaut attribuée par le système
# est élevée (30), ce qui peut empêcher le choix des routes
# pour cette interface.
#
# Valeurs possibles : auto, système, < toute valeur entière
# positive>.
#
# - auto : La valeur de la métrique est choisie par freelan
# afin que le réseau VPN ait priorité sur le réseau physique.
# - système : La valeur par défaut du système est prise (généralement 30).
# Toute valeur entière positive (ex. 3).
#
# Valeur par défaut: auto
#metric=auto

# L'adresse IPv4 de l'adaptateur de prise et la longueur du préfixe à utiliser.
#
# L'adresse réseau doit être au format numérique avec un suffixe netmask.
#
# Sous Windows, la longueur du préfixe est ignorée (mais doit quand même être spécifiée) et le masque de réseau est déterminé en fonction de la classe IPv4. Il est recommandé de
```

```
définir l'option network.enable_dhcp_proxy.  
#  
# En commentant, il n'y aura pas de mise en réseau IPv4. Vous  
ne pouvez pas fournir de valeur à blanc.  
#  
ipv4_address_prefix_length=9.0.0.1/24  
  
# L'adresse IPv6 de l'adaptateur de prise et la longueur du pr  
éfixe à utiliser.  
#  
# L'adresse réseau doit être au format numérique avec un  
suffixe netmask.  
#  
# En commentant, il n'y aura pas de mise en réseau IPv6. Vous  
ne pouvez pas fournir de valeur à blanc.  
#  
ipv6_address_prefix_length=2aa1::1/8  
  
# L'adresse IPv4 distante pour les interfaces tun.  
#  
# Certains systèmes utilisent cette adresse combinée avec la  
longueur de préfixe spécifiée dans '  
ipv4_address_prefix_length' pour créer la route qui utilise  
l'adaptateur tun.  
#  
# S'il n'est pas spécifié, l'adresse par défaut est l'adresse  
réseau associée à 'ipv4_address_prefix_length'.  
#  
# Par exemple, si 'ipv4_address_prefix_length' est  
"9.0.1.5/24", alors la valeur par défaut de '  
remote_ip4_address' sera "9.0.1.0".  
#  
# Ce paramètre est ignoré en mode tap.  
#  
# Par défaut : <network address of 'ipv4_address_prefix_length  
'>  
#remote_ip4_address=9.0.0.0  
  
# S'il faut activer le proxy ARP.  
#
```

```
# Lorsque le proxy ARP est activé, toutes les requêtes ARP  
# sont redirigées silencieusement vers un serveur ARP interne  
# qui répond toujours positivement. Les hôtes distants ne re-  
#çoivent jamais de requête ARP.  
#  
# Warning : Le réglage de ce paramètre peut entraîner des  
# problèmes de connectivité. Il est fourni uniquement à des  
# fins de débogage et de test.  
#  
# Valeur par défaut: no  
#arp_proxy_enabled=no  
  
# La fausse adresse Ethernet du proxy ARP.  
#  
# Si tap_adapter.arp_proxy_enabled n'est pas défini, cette  
# option est ignorée.  
#  
# Par défaut : 00:aa:bb:cc:dd:ee  
#arp_proxy_fake_ethernet_address=00:aa:bb:cc:dd:ee  
  
# S'il faut activer le proxy DHCP.  
#  
# Lorsque le proxy DHCP est activé, toutes les requêtes BOOTP/  
# DHCP sont redirigées en silence vers un serveur DHCP  
# interne. Les hôtes distants ne reçoivent jamais de requête  
# DHCP.  
#  
# L'interface TAP doit être prête à émettre des requêtes DHCP  
# si cette option est activée.  
#  
# L'utilisation de cette option est utile principalement sur  
# les anciennes versions de Windows pour les adresses IPv4.  
# Sous Windows, si cette option est activée, aucune tentative  
# de réglage direct de l'adresse IPv4 ne sera effectuée.  
#  
# Valeur par défaut : yes  
#dhcp_proxy_enabled=yes  
  
# L'adresse IPv4 du serveur proxy DHCP et la longueur du pré-  
# fixe à utiliser.
```

```
#  
# Cette valeur doit être différente de network.  
#      ipv4_address_prefix_length mais dans le même réseau.  
#  
# Notez que bien que cette option attend une adresse IPv4  
# valide de l'hôte, fournir une adresse réseau fonctionne également sur les systèmes d'exploitation Windows et POSIX.  
#  
# Valeur par défaut: 9.0.0.0/24  
#dhcp_server_ipv4_address_prefix_length=9.0.0.0/24  
  
# L'adresse IPv6 du serveur proxy DHCP et la longueur du préfixe à utiliser.  
#  
# Cette valeur doit être différente de network.  
#      ipv6_address_prefix_length mais dans le même réseau.  
#  
# Notez que bien que cette option attend une adresse IPv6 valide de l'hôte, fournir une adresse réseau fonctionne également sur les systèmes d'exploitation Windows et POSIX.  
#  
# Valeur par défaut: 2aa1::/8  
#dhcp_server_ipv6_address_prefix_length=2aa1::/8  
  
# Le script à appeler lorsque l'adaptateur de prise est en marche.  
#  
# Le script est appelé avec le nom de l'adaptateur de prise comme premier argument.  
#  
# L'état de sortie du script est ignoré.  
#  
# Par défaut: <empty>  
#up_script=  
  
# Le script à appeler lorsque l'adaptateur de prise est posé.  
#  
# Le script est appelé avec le nom de l'adaptateur de prise comme premier argument.  
#
```

```
# L'état de sortie du script est ignoré.  
#  
# Par défaut: <empty>  
#down_script=  
  
[switch]  
  
# La méthode de routage des messages.  
#  
# Valeurs possibles : commutateur, concentrateur  
#  
# - Switch : Agir comme un interrupteur. Les messages ne sont  
#   envoyés au bon hôte que lorsque son adresse est connue.  
# - moyeu : Envoyez tous les messages à tous les membres du ré  
#   seau. L'empreinte mémoire est légèrement réduite au prix d'  
#   une utilisation beaucoup plus importante de la bande  
#   passante. Recommandé pour  
# Réseaux 1 contre 1 uniquement.  
#  
# Warning : A tout moment, si la consommation de mémoire est  
#   trop élevée, la méthode routing_method peut être  
#   temporairement commutée de "switch" à "hub" pour éviter  
# DoS attacks.  
#  
# Par défaut: switch  
#routing_method=switch  
  
# Activer ou non le mode relais.  
#  
# Valeurs possibles : non, oui  
#  
# Non : ne pas relayer les images d'un hôte distant à l'autre.  
# Oui : Transmet des images d'un hôte à l'autre.  
#  
# Si vous activez le mode relais, il est recommandé que  
#   routing_method soit réglé sur switch.  
#  
# Default: no  
#relay_mode_enabled=no
```

```
[router]
```

```
# Les routes IP locales.  
#  
# La liste des itinéraires à annoncer aux autres pairs.  
#  
# Ces routes peuvent contenir une passerelle.  
#  
# Vous pouvez répéter l'option local_ip_route pour ajouter  
# plusieurs routes.  
#  
# Exemples :  
# - 192.168.0.0/24  
# - 192.168.0.0/24 => 9.0.0.1  
# - fe80::1/64  
# - fe80::1/64 => fe80::::fffff  
# - 0.0.0.0/0  
# - 0.0.0.0/0 => 9.0.0.1  
# - ::/0  
# - ::/0 => fe80::::fffff  
# - ipv4_proxy  
# - ipv6_proxy  
#  
# 'ipv4_proxy' et 'ipv6_proxy' sont des valeurs spéciales é  
# quivalentes a '0.0.0.0.0/0 => <tap_adapter.ipv4_address>'  
# et '::/0 => <tap_adapter.ipv6_address>' .  
#  
# Ces instructions sont particulièrement utiles lors de la  
# configuration d'un proxy VPN.  
#  
# Par défautt: <none>  
#local_ip_route=192.168.0.0/24  
  
# Les serveurs DNS locaux.  
#  
# La liste des serveurs DNS à annoncer aux autres pairs.  
#  
# Vous pouvez répéter l'option local_dns_server pour ajouter  
# plusieurs serveurs DNS.  
#
```

```
# Exemples :
# - 8.8.8.8
# - 2001:4860:4860::8888
#
# Par défaut: <none>
#local_dns_server=192.168.0.254

# Activer ou non le routage client.
#
# Valeurs possibles : non, oui
#
# Non : Ne pas relayer les trames IP d'un hôte distant à l'autre.
# Oui : Agir comme routeur IP et relayer les trames IP d'un hôte à l'autre.
#
#
# Valeur par défaut: yes
#client_routing_enabled=yes

# Accepter ou rejeter les demandes d'acheminement provenant d'autres pairs.
#
# Désactiver cette option en mode tun causera des problèmes de connectivité.
#
# Valeur par défaut: yes
#accept_routes_requests=yes

# La politique interne d'acceptation des itinéraires.
#
# Indique le type de routes à accepter d'autres hôtes.
#
# Les routes seront utilisées en interne.
#
# Valeurs possibles : none, unicast_in_network, unicast, subnet, any
#
# - aucun : Je n'accepte aucun itinéraire. Utilisez cette option pour désactiver la fonction.
```

```
# - unicast_in_network : N'accepter que les routes unicast qui
#   appartiennent au réseau de l'interface locale.
# - unicast : N'acceptez que les routes unicast.
# - sous-réseau : N'accepter que les routes du sous-réseau qui
#   appartiennent au réseau de l'interface locale.
# - n'importe lequel : Acceptez n'importe quel itinéraire.
#
# Remarque : cette option est ignorée en mode tap, car tap ne
#   fait pas de routage IP interne.
#
# Par défaut: unicast_in_network
#internal_route_acceptance_policy=unicast_in_network

# Le système achemine la politique d'acceptation.
#
# Indique le type de routes à accepter d'autres hôtes.
#
# Ces itinéraires seront ajoutés à la table de routage du syst
# ème.
#
# Valeurs possibles : none, unicast, any, unicast_with_gateway
#   , any_with_gateway, any_with_gateway
#
# - aucun : Je n'accepte aucun itinéraire. Utilisez cette
#   option pour désactiver la fonction.
# - unicast : N'acceptez que les routes unicast. Ceux qui
#   contiennent une passerelle sont rejetés.
# - n'importe lequel : Acceptez n'importe quel itinéraire sauf
#   ceux qui contiennent une passerelle.
# - unicast_with_gateway : N'acceptez que les routes unicast,
#   même celles qui contiennent une passerelle.
# any_with_gateway : Acceptez n'importe quel itinéraire, même
#   ceux qui contiennent une passerelle.
#
# Les routes qui appartiennent au réseau d'interface actuel
#   sont ignorées silencieusement car la table de routage du
#   système les contient déjà.
#
# Remarque : cette option est active en mode tun et tap car
#   elle affecte la table de routage du système.
```

```
#  
# Note 2 : En mode tun, les routes sont d'abord filtrées par  
# la politique interne_route_acceptance_policy.  
#  
# Attention : assurez-vous de bien comprendre les implications  
# que cette option peut avoir. Autoriser des modifications  
# de la table de routage du système pour d'autres hôtes peut  
# représenter un risque énorme pour la sécurité.  
#  
# Valeur par défaut: none  
#system_route_acceptance_policy=none  
  
# Le nombre maximum de routes à accepter pour un hôte donné.  
#  
# Valeurs possibles : 0, <a nombre positif>> 0  
#  
# - 0 : Aucune limite.  
<un nombre positif> : Seul un nombre fini d'itinéraires est  
# accepté à partir d'autres hôtes.  
#  
# Remarque : la limite est appliquée séparément aux adresses  
# IPv4 et IPv6. Ce qui signifie qu'une limite de 1 permet d'  
# avoir une adresse de chaque type.  
#  
# Valeur par défaut: 1  
#maximum_routes_limit=1  
  
# La politique d'acceptation des serveurs DNS.  
#  
# Indique le type d'adresses de serveur DNS à accepter d'  
# autres hôtes.  
#  
# Valeurs possibles : none, in_network, any  
#  
# - aucun : N'accepte aucun serveur DNS. Utilisez cette option  
# pour désactiver la fonction.  
# - dans_réseau : N'acceptez que les adresses de serveur DNS  
# qui appartiennent au réseau IP de l'interface.  
# - n'importe lequel : Accepter n'importe quelle adresse de  
# serveur DNS.
```

```
#  
# Valeur par défaut: in_network  
#dns_servers_acceptance_policy=in_network  
  
# Le script à appeler lorsqu'une entrée DNS doit être ajoutée  
# ou supprimée.  
#  
# Le script est appelé avec le nom de l'adaptateur de prise  
# comme premier argument.  
# Le deuxième argument est un verbe qui peut être :  
# - ajouter : Une entrée DNS doit être ajoutée.  
# - enlevez : Une entrée DNS doit être supprimée.  
# Le troisième argument est l'adresse du serveur DNS à ajouter  
# ou à supprimer.  
#  
# Si le script sort avec une valeur non nulle, on suppose que  
# l'ajout ou la suppression de l'entrée DNS a échoué. Si l'  
# ajout échoue pour une adresse donnée, le script ne sera pas  
# appelé à être supprimé pour cette même adresse.  
#  
# Sous Windows, si aucun script n'est fourni, FreeLAN ajoutera  
# ou supprimera le serveur DNS en utilisant les appels système.  
#  
# Sous Mac OS X et Linux, il n'y a malheureusement pas d'appel  
# système fiable et vous DEVEZ fournir un script ou les  
# paramètres DNS seront simplement ignorés.  
#  
# Par défaut: <empty>  
#dns_script=  
  
[security]  
  
# La phrase de chiffrement utilisée pour générer une clé pré-  
# partagée à utiliser pour le chiffrement.  
#  
# La PSK est dérivée à l'aide de PBKDF2.  
#  
# L'utilisation d'un PSK est moins sûre que l'utilisation d'un  
# certificat et ne devrait jamais être un premier choix. Il
```

```
est utile dans les cas où il n'est pas possible de générer
des certificats ou des clés privées.

#
# Vous pouvez spécifier un PSK même si vous avez un certificat
# , ce qui permet de vous connecter avec des noeuds sans
# certificat.

#
# La phrase de passe DOIT rester secrète.

#
# Par défaut: <none>
#passphrase=

# Le sel à utiliser pour dériver le PSK de la phrase de
# chiffrement.

#
# Il est recommandé de changer cette valeur pour votre propre
# installation freelan lorsque vous utilisez des PSK. Il n'
# est pas nécessaire que ce soit un secret, mais il devrait
# idéalement être unique.

#
# Par défaut : freelan
#passphrase_salt=freelan

# Le nombre d'itérations à utiliser pour dériver le PSK de la
# phrase de chiffrement.

#
# Vous pouvez augmenter (ou diminuer, mais s'il vous plaît, ne
# le faites pas) ce nombre pour augmenter le temps qu'il
# faut pour dériver la clé de la phrase de passe et réduire
# la probabilité d'attaques par force brute.

#
# Valeur par défaut: 2000
#passphrase_iterations_count=2000

# Le sel à utiliser pour dériver le PSK de la phrase de
# chiffrement.

#
# Par défaut: freelan
#passphrase_salt=freelan
```

```
# Le fichier de certificat X509 à utiliser pour la signature.  
#  
# A moins que client.enabled ne soit réglé sur "yes" ou qu'un  
# PSK ne soit spécifié, ce paramètre est obligatoire.  
#  
# Par défaut : <none>  
#signature_certificate_file=  
  
# Le fichier de clé privée à utiliser pour la signature.  
#  
# Ce paramètre est obligatoire, sauf si le paramètre client.  
# enabled a la valeur "yes" ou si PSK est spécifié.  
#  
# Cette clé privée doit correspondre au fichier de certificat  
# de signature spécifié.  
#  
# Par défaut : <none>  
#signature_private_key_file=  
  
# La méthode de validation de certificat à utiliser.  
#  
# Les valeurs possibles sont : par défaut, aucune  
#  
# - par défaut : Correspond à n'importe quel certificat pré  
#   senté par rapport aux autorités de certification spécifiées  
#  
# - aucun : Désactiver la validation des certificats.  
#  
# Warning : Réfléchissez à deux fois avant de régler "none"  
# ici, car cela désactive complètement la validation du  
# certificat. Si vous choisissez de le faire, assurez-vous d'  
# avoir un script de validation de certificat robuste défini  
# comme certificate_validation_script.  
#  
# Par défaut : default  
#certificate_validation_method=default  
  
# Le script de validation de certificat à appeler.  
#
```

```
# Chaque fois qu'un certificat externe est reçu et accepté par
# la méthode certificate_validation_method spécifiée, le
# script spécifié est appelé avec un nom de fichier
# certificat X509 comme premier argument.
#
# Si l'état de sortie du script est zéro, le certificat est
# accepté.
# Si l'état de sortie du script est différent de zéro, le
# certificat est rejeté.
#
# Le script de validation de certificat est appelé même si la
# méthode certificate_validation_method est définie sur "none"
# .
#
# Spécifiez un chemin de script de validation vide pour dé
# sactiver la validation de script.
#
# Par défaut: <empty>
#certificate_validation_script=

# Les certificats d'autorité.
#
# Vous pouvez répéter l'option authority_certificate_file pour
# spécifier plusieurs certificats d'autorité.
#
# Par défaut : <none> <none>
#Fichier_certificat_d'autorité=

# La méthode de validation de la révocation des certificats à
# utiliser.
#
# Les valeurs possibles sont : last, all, none
#
# - dernier : Seul le dernier certificat de la chaîne de
# certification est vérifié pour la révocation.
# - tous : Tous les certificats de la chaîne de certification
# font l'objet d'un contrôle de révocation.
# - aucun : La révocation des certificats n'est pas vérifiée.
#
# Valeur par défaut: none
```

```
#certificate_revocation_validation_method=none

# Les listes de révocation des certificats.
#
# Vous pouvez répéter l'option
# certificate_revocation_list_list_file pour spécifier
# plusieurs listes de révocation de certificats.
#
# Par défaut: <none>
#certificate_revocation_list_file=
```

9.2.6 Fichier de configuration de Freelan

Le fichier suivant est le fichier de configuration de freelan qui est localisé dans /etc/freelan/freelan.cfg . Pour que ça fonctionne il faut rajouter les certificats qui permettent de se connecter au serveur.

```
[fscp]
listen_on=0.0.0.0:443
contact=10.105.17.1:443
cipher_capability=aes256-gcm

[tap_adapter]
ipv4_address_prefix_length=9.0.0.3/24
dhcp_proxy_enabled=yes
dhcp_server_ipv4_address_prefix_length=9.0.0.0/24

[security]
signature_certificate_file="/freelan/hv2.crt"
signature_private_key_file="/freelan/hv2.key"
authority_certificate_file="/freelan/ca.crt"
```

9.2.7 Docker freelan

Pour créer une image docker de freelan sur magiea 6 voici le dockerfile :

```
FROM mageia:latest
```

```

ENV DEPENDENCIES boost-devel libcurl-devel openssl-devel
      miniupnpc-devel git easypmbuilder scons

ENV FREELAN_BRANCH master

# Install dependencies
RUN dnf update && \
dnf install -y $DEPENDENCIES

# Get FreeLAN sources and compile it
RUN git clone -b $FREELAN_BRANCH --depth=100 https://github.com/freelan-developers/freelan.git && \
cd freelan && \
scons --mode=release install prefix=/usr/ -j2 && \
cd .. && rm -rf freelan

# Profit !

EXPOSE 12000/udp

```

9.3 FirewallD

9.3.1 Installation

Sur certaines distributions, Firewalld est installé nativement, mais pour Mageia il faut l'installer :

```
# dnf install firewalld
```

Firewalld étant un service il est possible de le stopper et de le démarrer avec la commande systemctl :

```
# systemctl stop firewalld
# systemctl start firewalld
# systemctl restart firewalld
```

9.3.2 Lancement de Firewalld au démarrage

Au démarrage de l'OS Firewalld est bloqué par Shorewall, c'est pourquoi nous avons créé un script qui désactive Shorewall et active Firewalld. Notre script s'appelle firewalld

```
#!/bin/bash

systemctl stop shorewall
systemctl disable shorewall
systemctl start firewalld
```

FIGURE 9.1 – script_firewalld

```
#!/bin/sh

/home/firewalld
```

FIGURE 9.2 – rc.local

et est dans /home (cf figure script_firewalld).

Une fois créé nous le lançons au démarrage. Pour cela, il faut créer le script /etc/rc.d/rc.local (cf figure rc.local). /home/firewalld étant la localisation du script précédent.

Chapitre 10

Sécurisation Docker

- Utilisation des cgroups afin de limiter l'utilisation de la mémoire, du cpu, du disk ...
Bien vérifier que la somme des ressources allouées aux conteneurs ne dépasse pas la capacité maximale de la machine hôte (en mémoire, en cpu, en disk :))
- Monter les fichiers dont nous avons réellement besoin. Il est préférable de les monter en "read only" ... Soit par lien symbolique. Soit en passant par un volume partagé.
- Ne jamais lancer un conteneur en root (-privileged à bannir)
- Utiliser les namespace. Ils partitionnent les ressources du noyau de sorte qu'un ensemble de processus voit un ensemble de ressources tandis qu'un autre ensemble de processus voit un autre ensemble de ressources. C'est la base de la conteneurisation sur Linux.

Quelques éléments propres à la sécurisation d'un Linux.

- Utiliser les Seccomp pour autoriser ou non certains appels système par défaut docker autorise 44 appels système. Possibilité de créer des profils qui interdisent/ autorisent tel ou tel appel système.
- Mettre votre image sur le hub docker afin que ce dernier puisse la scanner avec son nouvel outil le "Docker security scaning" <https://www.developpez.com/actu/98661/Docker-devoile-Docker-Security-Scanning-son-outil-de-scan-de-vulnerabilites-au-sein-des-containers-Docker/>
- lien utile : <https://www.youtube.com/watch?v=sK5i-N34im8>

Voici la liste des technologies qui permettent de sécurisé Docker :

- AppArmor " est un logiciel libre de sécurité pour Linux. AppArmor permet à l'administrateur système d'associer à chaque programme un profil de sécurité qui restreint les capacités de celui-ci.

Il s'agit plus précisément d'un outil qui permet de verrouiller les applications en limitant strictement leur accès aux seules ressources auxquelles elles ont droit sans perturber leur fonctionnement. "¹.

- SELinux " est un Linux security module (LSM), qui permet de définir une politique de contrôle d'accès obligatoire aux éléments d'un système issu de Linux."².
- grsecurity est une modification augmentant la sécurité pour le noyau Linux distribué sous la licence publique générale GNU version 2. Il inclut différents éléments, dont PaX, un système de contrôle d'accès à base de rôles et différents moyens de renforcer la sécurité générale du noyau.³.
- Voici le lien d'un site qui référence les bonnes pratiques : <https://learn.cisecurity.org/benchmarks> voici un outil créé par la communauté qui permet de vérifier les spécifications de CIS. <https://github.com/docker/docker-bench-security> Les résultats sont intéressants à observer.

Dans notre cas, nous allons étudier deux solutions pour notre projet. Dans un premier temps nous étudierons SELinux car il est commun aux distributions de la famille des redhat. Nous étudierons aussi Msec qui est un outil de sécurisation natif à Mageia. Pour un souci de facilité nous ne verrons pas AppArmor qui est plutôt utilisé pour les linux issus de la famille de debian. Tandis que grsecurity n'est pas packager sur Mageia.

10.1 SELinux

10.1.1 Approche SELinux pour la sécurisation Linux

Dans notre projet nous utilisons une distribution Mageia 6. Comme nous pouvons le voir sur : Arborescence des forks linux coté Mageia.La distribution mageia est un fork de RedHat. Il existe bien la commande : **urpmi selinux-policy** pour l'installation. Mais actuellement Mageia :6 n'inclut pas SELinux dans son noyau. Et d'après les développeurs de chez mageia ils ne l'incluront pas dans la nouvelle version.

Le système SELinux a été créé par la NSA. SELinux se base sur une approche de sécurité de type TE (Type Enforcement). Les objets (fichier, processus) ont une étiquette, c'est le "type" pour les fichiers et le "domaine pour les processus. L'objectif est de définir des interactions permises ou non entre domaine et type. Le concurrent de SELinux est Apparmor.

Liste de liens utiles :

-
1. <https://doc.ubuntu-fr.org/apparmor>
 2. <https://fr.wikipedia.org/wiki/SELinux>
 3. <https://fr.wikipedia.org/wiki/Grsecurity>

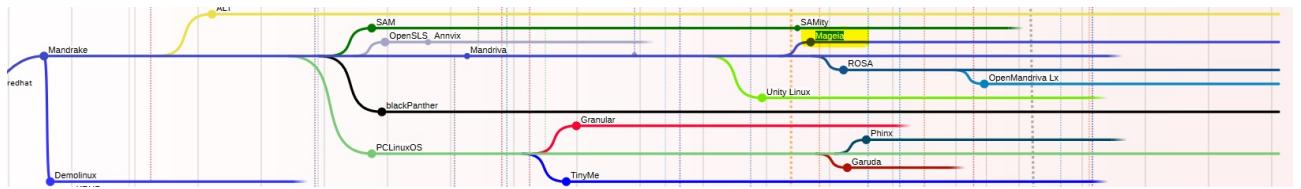


FIGURE 10.1 – Arborescence des forks linux côté Mageia

- Le site officiel <https://www.nsa.gov/what-we-do/research/selinux/>
- Pour comprendre SELinux : <https://wiki.gentoo.org/wiki/SELinux/Tutorials>
<https://doc.fedoraproject.org/wiki/SELinux>
- Le livre : SELinux Cookbook

10.1.2 Les SC (Security Context)

Un SC est associé aux différents objets du système. Les objets sont par exemple : fichiers, processus, IPC, sockets ...Les SC sont composés trois parties :

- L'identité : Nom du propriétaire de l'objet, par défaut un utilisateur à l'identité user_u.
- Le rôle : Les rôles sont l'intermédiaire entre les utilisateurs et les domaines SELinux. Les domaines sont accessibles par des rôles définis, et les rôles sont eux-mêmes accessibles par des utilisateurs définis.
- Le type : Un type est un regroupement d'objets sur leur similarité du point de vue de la sécurité.

10.1.3 Les booléens

Les booléens sont l'ensemble des règles utilisées par SELinux. Elles permettent une configuration simplifiée de SELinux. Il est possible d'installer une interface graphique pour configurer SELinux

```
% en root
dnf install policycoreutils-gui
```

10.1.4 Les commandes utiles

afficher le statut actuel de SELinux ;

```
sestatus
```

obtenir le mode SELinux courant de votre machine ;

`getenforce`

afficher la liste des fichiers et dossiers ainsi que leur contexte SELinux ;

`ls -Z`

La même chose mais pour les processus ;

`ps -Z`

Information sur les booléens ;

`getsebool`

modifier le mode SELinux de votre machine ;

`setenforce`

modifier la valeur d'un booléen ;

`setsebool`

modifier le contexte SELinux d'un fichier ;

`chcon`

gérer les politiques SELinux.

`semanage`

10.2 Portsentry

Nous pouvons utiliser **portsentry** pour bloquer le scan de ports. Pour cela nous devons l'installer. Et changer le fichier de configue.

```
$ sudo dnf install install portsentry  
$ sudo nano /etc/portsentry/portsentry.conf
```

Commentez les lignes

`KILL_HOSTS_DENY`

Décommentez la ligne

```
KILL_ROUTE="/sbin/iptables -I INPUT -s $TARGET$ -j DROP".
```

Démarrer **portsentry**

```
$ sudo portsentry -audp  
$ sudo portsentry -atcp
```

Lien vers le wiki de **portsentry** <https://wiki.debian-fr.xyz/Portsentry>

10.3 Fail2ban

Fail2ban lit des fichiers de log et bannit les adresses IP qui ont obtenu un trop grand nombre d'échecs lors de l'authentification. Pour cela, Fail2ban va mettre à jour les règles du pare-feu pour rejeter les adresses IP qui ont obtenu un trop grand nombre d'échec. Fail2ban peut également lire directement dans les fichiers log du serveur Apache par exemple.

Pour l'installation et la configuration nous conseillons cette documentation : <https://wiki.debian-fr.xyz/Fail2ban>

10.4 rkhunter

rkhunter (pour Rootkit Hunter) est un programme Unix qui permet de détecter les rootkits, portes dérobées et exploits. Léger, il analysera le système une fois par jour par défaut. Voici le lien de configuration : <https://wiki.debian-fr.xyz/Rkhunter>

10.5 Prelude

Prelude est un système universel de (SIEM). Prelude collecte, normalise, trie, agrège, met en corrélation et rapporte tous les événements liés à la sécurité indépendamment de la marque du produit ou de la licence donnant lieu à ces événements

En plus d'être capable de récupérer tout type de logs (logs système, syslog, fichiers plats, etc.), Prelude bénéficie d'un support natif avec de nombreux systèmes dédiés afin d'enrichir ses informations (snort, samhain, ossec, auditd, etc.).

Pour plus d'informations : <https://www.prelude-siem.org/projects/prelude/wiki/ManualUser>

10.6 Surveiller les logs

Il est conseillé de réaliser un script qui effectue des greps sur les fichiers de log qui sont intéressants du point de vue de la sécurité. Liste des fichiers à surveiller :

- /var/log/auth.log commande utile : cat /var/log/auth.log | grep authentification failure. Pour afficher les authentifications qui ont échoué.
- /var/log/messages Ce journal doit être considéré comme le journal des "activités générales du système".
- /var/log/syslog enregistre tout, sauf les messages relatifs aux authentifications.
- /var/log/fail2ban commande utile :at /var/log/fail2ban | grep ban. Pour afficher la liste des bannis par Fail2ban
- /var/log/rkhunter

Pour faciliter le travail de lecture de ces fichiers log il existe **logwatch**. Voici le lien pour l'installation et le configurer. <https://wiki.debian-fr.xyz/Logwatch>

10.7 Côté Linux

- Vérifier les services et le deamons qui se lance au démarrage.
- Editer le fichier /etc/login.defs.
- Utiliser pwunconv puis pwconv afin d'établir une durée de vie à l'utilisateur.
- Recréer un groupe discret qui aura les droits privilégiés.
- Fichiers .rhosts et hosts.equiv. Il faut bloquer la recréation de ces fichiers, avec les commandes :

```
touch /.rhosts /etc/hosts.equiv  
chmod 0 /.rhosts /etc/hosts.equiv
```

- Protéger les fichiers systèmes de l'écriture

chattr

Utilisation de mots de passe forts. Utiliser textbfPAN lien utile : <https://www.it-connect.fr/gestion-de-la-politique-des-mots-de-passe-sous-linux/> ou générer un mot de passe avec pwgen dans certains cas de figure.

Chapitre 11

Documentation utilisation utilisateur

Nous n'avons pas encore rédigé les documentations utilisateurs, cependant elles arriveront au second semestre.

Chapitre 12

Les tests réalisés

12.1 LXC

LXC (Linux Containers)- Pourquoi ça ne fonctionne pas ?

12.1.1 Installation :

```
dnf install lxc
```

S'assurer que nous disposons des fichiers /etc/subuid et /etc/subgid. Dans le cas contraire les créer. Dans notre cas, cf figure lxc-sub. On doit les créer.

```
touch subuid  
touch subguid
```

On crée maintenant un utilisateur LXC :

```
useradd user_lxc
```

On s'assure que l'utilisateur a un subuid et un subgid dans /etc/subuid et /etc/subgid avec les commandes :

```
[root@localhost etc]# cat /etc/subuid  
cat: /etc/subuid: Aucun fichier ou dossier de ce type  
[root@localhost etc]# cat /etc/subgid  
cat: /etc/subgid: Aucun fichier ou dossier de ce type
```

FIGURE 12.1 – lxc-sub

```
[root@localhost user]# lxc-create -t download -n my-container
Setting up the GPG keyring
ERROR: Unable to fetch GPG key from keyserver.
lxc-create: lxccontainer.c: create run template: 1295 container creation template for my-container failed
lxc-create: tools/lxc_create.c: main: 318 Error creating container my-container
[root@localhost user]#
```

FIGURE 12.2 – test_root

```
cat /etc/subuid
cat /etc/subgid
```

Il faut ensuite aller dans /etc/lxc/lxc-usernet pour autoriser l'utilisateur sans privilège à créer des conteneurs. On ajoute la ligne suivante :

```
user_lxc veth lxcbr0 10
```

Dans le fichier que l'on crée :

```
nano lxc-usernet
```

On crée ensuite le dossier /.config/lxc :

```
cd ~/.config
mkdir lxc
```

On copie le contenu de /etc/lxc/default.conf dans /.config/lxc/default.conf

```
cp /etc/lxc/default.conf ~/.config/lxc/default.conf
```

On y ajoute les lignes suivantes :

```
lxc.idmap = u 0 100000 65536
lxc.idmap = g 0 100000 65536
```

12.1.2 Tentative en root

(cf figure test_root). Cela est dû au proxy de l'école, pour contourner le problème :

```
lxc-create -t download -n -my-container \
-- --keyserver hkp://p80.pool.sks-keyserver.net:80
```

Cela fonctionne cf figure test_lxc1 et test_lxc2.

12.1.3 Tentative en utilisateur lxc

(cf figure test_user_lxc)

```
[root@localhost user]# lxc-create -t download -n test -- --keyserver hkp://p80.pool.sks-keyservers.net:80
Setting up the GPG keyring
Downloading the image index
...
DIST      RELEASE ARCH      VARIANT BUILD
...
alpine  3.4      amd64  default 20180627_17:50
alpine  3.4      armhf  default 20180627_17:50
alpine  3.4      i386   default 20180627_17:50
alpine  3.5      amd64  default 20181101_13:00
alpine  3.5      armhf  default 20181101_13:02
alpine  3.5      i386   default 20181101_13:03
alpine  3.5      amd64  default 20181101_13:00
alpine  3.6      amd64  default 20181101_13:00
alpine  3.6      armhf  default 20181101_13:02
alpine  3.6      i386   default 20181101_13:02
alpine  3.6      amd64  default 20181101_13:02
alpine  3.6      armhf  default 20181101_13:02
alpine  3.6      i386   default 20181101_13:01
alpine  3.7      amd64  default 20181101_13:00
alpine  3.7      armhf  default 20181101_13:03
alpine  3.7      i386   default 20181101_13:02
alpine  3.7      amd64  default 20181101_13:01
alpine  3.7      armhf  default 20181101_13:01
alpine  3.7      i386   default 20181101_13:00
...
I
```

FIGURE 12.3 – test_lxc1

```
ubuntu trusty i386  default 20181101_07:42
ubuntu trusty powerpc default 20180824_07:43
ubuntu trusty ppc64el default 20181101_07:43
ubuntu xenial amd64  default 20181101_07:42
ubuntu xenial arm64  default 20181101_07:45
ubuntu xenial armhf default 20181101_07:44
ubuntu xenial i386   default 20181101_07:42
ubuntu xenial s390x default 20180824_07:44
ubuntu xenial s390xpc default 20180824_07:44
ubuntu xenial ppc64el default 20181101_07:43
ubuntu xenial s390x  default 20181101_07:43
...
I
```

FIGURE 12.4 – test_lxc2

```
[user_lxc@localhost user]$ lxc-create -t download -n my-container \
> -- --keyserver hkp://p80.pool.sks-keyservers.net:80
lxc-create: utils.c: mkdir_p: 253 Permission denied - failed to create directory '/run/user/1000/lxc/'
Failed to create lock
Failed to create lxc container.
```

FIGURE 12.5 – test_user_lxc

```
lxc-cls: lxccontainer.c: lxc container new: 4149 Error: test creation was not completed
NAME      STATE  AUTOSTART GROUPS IPV4 IPV6
Nom      STOPPED 0   :   :   :
my-container STOPPED 0   :   :   :
```

FIGURE 12.6 – lxc-ls

```
[user_lxc@localhost ~]$ sudo urpmi --auto-update
[sudo] Mot de passe de user_lxc :
user_lxc n'apparaît pas dans le fichier sudoers. Cet événement sera signalé.
[user_lxc@localhost ~]$
```

FIGURE 12.7 – urpmi

On ne dispose pas des droits alors qu'on le devrait à ce point. On réécrit les règles en tant qu'utilisateur.

Après plusieurs tentatives : Mettre dans le fichier `/.config/lxc/default.conf` en tant qu'utilisateur :

```
lxc.id_map = u 0 100000 65536
lxc.id_map = g 0 100000 65536
lxc.network.type = veth
lxc.network.link = lxcbr0
```

En tant que root :

```
unset la variable XDG_SESSION_ID XDG_RUNTIME_DIR
```

On peut alors créer en tant qu'utilisateur un conteneur :

```
lxc-creat -t download -n -my-container \
-- --keyserver hkp://p80.pool.sks-keyservers.net:80
```

Voyons voir si cela a fonctionné avec la commande :

```
lxc-ls -f
```

résultat cf figure lxc-ls

on voit bien que l'utilisateur ne possède pas les droits root cf figure urpmi.

Launchons le conteneur :

```
lxc-start -n my-container -d -F
```

On obtient une erreur concernant les Cgroups (cf figure Cgroups).

Après quelques recherches, il s'avère que les cgroups sous Mageia sont gérés via le fichier selinux.

"Right, I forgot to comment on this one. Fedora 27 comes with selinux enabled by default and a specific profile. All cgroups mounts in the cgroup filesystem carry a seclabel option"

```
[user@localhost ~]# lxc-start -n my-container -d F
lxc-start: cgroups/cgfs.c: lxc_cgroufs_create: 999 Could not set clone_children to 1 for cpuset hierarchy in parent cgroup.
lxc-start: cgroups/cgfs.c: lxc_cgroufs_create: 999 Could not set clone_children to 1 for cpuset hierarchy in parent cgroup.          lxc-start: cgro
ups/cgfs.c: cgroup_rmdir: 209 Read-only file system - cgroup_rmdir: failed to delete /sys/fs/cgroup/net_cls/
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Permission denied - cgroup_rmdir: failed to delete /sys/fs/cgroup/devices/user.slice
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Read-only file system - cgroup_rmdir: failed to delete /sys/fs/cgroup/btkie/
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Head-only file system - cgroup_rmdir: failed to delete /sys/fs/cgroup/cpuset/
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Read-only file system - cgroup_rmdir: failed to delete /sys/fs/cgroup/freezer/
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Head-only file system - cgroup_rmdir: failed to delete /sys/fs/cgroup/systemd/
cgroup/cpu,cpuacct/
lxc-start: cgroups/cgfs.c: cgroup_rmdir: 209 Permission denied - cgroup_rmdir: failed to delete /sys/fs/cgroup/systemd/
ser.slice/user-1001.slice/session-2.scope
lxc-start: start.c: lxc_spawn: 1119 Failed creating cgroup ps.
lxc-start: start.c: __lxc_start: 1354 Failed to spawn container "my-container".
lxc-start: tools/lxc_start.c: main: 366 The container failed to start.
lxc-start: tools/lxc_start.c: main: 378 Additional information can be obtained by setting the --logfile and --logpriority options.
```

FIGURE 12.8 – Cgroups

```
#!/bin/sh
echo 1 > /sys/fs/cgroup/cpuset/cgroup.clone_children
for cgroup in /sys/fs/cgroup/*; do
    mkdir -p ${cgrou
p}/user.slice/user-$({id -u ${1}}).slice
    chown -R ${id -u ${1}}:$({id -g ${1}}) ${cgrou
p}/user.slice/user-$({id -u ${1}}).slice

    if [ "$(basename ${cgrou
p})" != "unified" ]; then
        echo ${2} > ${cgrou
p}/user.slice/user-$({id -u ${1}}).slice/tasks
    fi
done
```

FIGURE 12.9 – lxc _ script

indicating that the filesystem supports selinux labels via xattrs. If you want to be able to run unprivileged container you need to take care to configure selinux correctly. As a starting point you should figure out what labels the cgroup mounts carry and what restrictions they bring. This is however very likely unrelated to liblxc itself so closing. In case the container needs to carry a specific selinux label you can use the lxc.selinux.context configuration key.
“

Installation du paquet selinux-policy-targeted – on obtient SELINUX

On le met en mode permissive :

```
/etc/selinux/ config SELINUX = permissive
```

Permet d'avoir des warnings. Cela ne change rien.

Nous avons besoin du package libpam-cfgs -> pas sous mageia : recherche équivalence, mais je ne l'ai pas trouvée.

```
chmod +x /home/lxc-usernet...
```

Au vu des logs on peut voir qu'il s'agit d'une erreur sur les cgroups. La solution que j'ai trouvée serait d'exécuter ce script, mais en tant que root (impossible pour notre projet) (cf figure script_lxc)

Il existe forcément une autre méthode et je pense que créer un cgroup serait possible.

```

lxc@localhost ~]$ lxc-checkconfig
--- Namespaces ---
Namespaces: enabled
Utsname namespace: enabled
Ipc namespace: enabled
Pid namespace: enabled
User namespace: enabled
Network namespace: enabled

--- Control groups ---
Cgroup: enabled
Group clone children flag: enabled
Cgroup device: enabled
Cgroup sched: enabled
Cgroup cpu account: enabled
Cgroup memory controller: missing
Cgroup cpuset: enabled

--- Misc ---
Veth pair device: enabled
Macvlan: enabled
Vlan: enabled
Bridges: enabled
Advanced netfilter: enabled
CONFIG_NF_NAT_IPV4: enabled
CONFIG_NF_NAT_IPV6: enabled
CONFIG_NP_NAT_TAMPER: enabled
CONFIG_IP6_NF_TARGET_MASQUERADE: enabled
CONFIG_IP6_NF_TARGET_HAVERSINE: enabled
CONFIG_NETFILTER_XT_TARGET_CHECKSUM: enabled
FUSE (for use with lxcfs): enabled

--- Checkpoint/Restore ---
File capabilities: enabled
CONFIG_FHANDLE: enabled
CONFIG_EVENTFD: enabled
CONFIG_EPOLL: enabled
CONFIG_UNIX_DIAG: enabled
CONFIG_NET_DIAG: enabled
CONFIG_PACKET_DIAG: enabled
CONFIG_NETLINK_DIAG: enabled
File capabilities: enabled

Note : Before booting a new kernel, you can check its configuration
Usage : CONFIG=/path/to/config /usr/bin/lxc-checkconfig

```

FIGURE 12.10 – config_lxc

Cependant, je n'ai pas les connaissances nécessaires pour le faire et je ne vais pas avoir suffisamment de temps. Il existe des docs sur les cgroups mais pas sur le problème que je rencontre et je ne sais pas quoi faire.

De plus, je pense qu'on va par la suite se frotter à un autre problème. En effet, lorsque l'on regarde la configuration de lxc :

lxc - checkconfig

(cf figure config_lxc) On remarque que le Cgroup memory controller est absent. Celui-ci est nécessaire au bon fonctionnement de lxc.

12.2 PODMAN

Vous trouverez dans ce document les tests que nous avons réalisés sur Podman. Nous allons vous expliquer l'intérêt que nous avons eu pour cette application. Pour comprendre le principe de conteneurisation, je vous renvoie vers l'état de l'art de ce rapport.

12.2.1 Podman vs Docker

Podman est dans la même philosophie que la ligne de commande de Docker. Podman permet également aux utilisateurs d'exécuter des groupes de conteneurs appelés pods. Un Pod est un terme développé pour le Projet Kubernetes qui décrit un objet qui a un ou

plusieurs processus conteneurisés partageant plusieurs espaces de noms (Réseau, IPC et éventuellement PID).

Podman apporte de l'innovation aux outils de conteneur dans l'esprit des commandes Unix qui font "une chose" bien. Podman n'a pas besoin d'un déamon pour faire fonctionner les conteneurs et les pods. Cela en fait un atout important pour votre arsenal d'outils de conteneur.

Au contraire, Docker de son côté utilise un "fat" déamon qui va gérer les containers. Docker est une solution propriétaire libre. Mais, du jour au lendemain Docker pourrait devenir payant. Podman est par conséquent plus léger par son fonctionnement.

La communauté Docker est plus importante de par son ancienneté. Docker est stable sur les différentes plateformes qui sont : Windows, Linux, Mac os. Aujourd'hui, Docker reste encore la solution de conteneurisation la plus répandue.

Une contrainte forte pour notre projet est l'utilisation de Mageia 6 mais ce dernier ne package pas Podman. En effet, Podman est une solution de conteneurisation en cours de développement, il n'existe qu'une version alpha. Voici le lien du projet : <https://github.com/containers/libpod>. Le leader du projet est : M Walsh sur le lien suivant, vous trouvez les informations de contact : <https://people.redhat.com/dwalsh/>.

12.2.2 Tests

Pour l'installation nous avons besoin des droits root.

```
$ su -
```

Dans notre cas, nous devons accéder aux sources Podman. Pour cela nous utilisons. Voir Figure 1.1.

```
dnf copr enable ngompa/containers-mga  
y
```

Nous procédons maintenant à l'installation de Podman avec son petit copain buildah (Figure 1.2).

```
dnf -y install Podman buildah
```

Ensuite, il suffit de télécharger une image.Dans notre cas, nous utilisons une image Mageia.

```
Podman pull mageia:6
```

Maintenant il suffit de lancer notre image (Figure 1.3). Mais comme nous pouvons le remarquer nous avons une erreur. La connection à notre conteneur est refusée.

```
[root@localhost ~]# dnf copr enable ngompa/containers-mga
You are about to enable a Copr repository. Please note that this
repository is not part of the main distribution, and quality may vary.

The Fedora Project does not exercise any power over the contents of
this repository beyond the rules outlined in the Copr FAQ at
<https://docs.pagure.org/copr.copr/user_documentation.html#what-i-can-build-in-copr>,
and packages are not held to any quality or security level.

Please do not file bug reports about these packages in Fedora
Bugzilla. In case of problems, contact the owner of this repository.

Do you want to continue? [y/N]: y
Activation du dépôt réussie.
[root@localhost ~]# _
```

FIGURE 12.11 – Activation des dépôts

```
Installé :
buildah.x86_64 1.4-4.git608fa84.mga6
podman.x86_64 0.10.1.3-4.gitdb08685.mga6
criu.x86_64 2.0-1.mga6
slirp4netns.x86_64 0.1-3.dev.git0037042.mga6
containernetworking-plugins.x86_64 0.7.3-3.mga6
containers-common.x86_64 0.1.32-3.dev.gite814f96.mga6
lib64nftnl4.x86_64 1.0.7-1.mga6
lib64ostree1.x86_64 2018.1-1.mga6
lib64protobuf-c1.x86_64 1.2.1-1.mga6
lib64seccomp2.x86_64 2.3.2-1.mga6
lib64selinux1.x86_64 2.5-6.mga6
nftables.x86_64 0.7-1.mga6
python-criu.x86_64 2.0-1.mga6
runc.x86_64 2:1.0.0-57.dev.git78ef28e.mga6

Terminé !
[root@localhost ~]#
```

FIGURE 12.12 – Installation

```
[root@localhost ~]# podman run mageia:6
error attaching to container c61f0bcd3f17540b3461cd74e37d693f6e30a99c8cdef20b882
dd96faa3e95ef: failed to connect to container's attach socket: /var/run/libpod/s
ocket/c61f0bcd3f17540b3461cd74e37d693f6e30a99c8cdef20b882dd96faa3e95ef/attach: d
ial unixpacket /var/run/libpod/socket/c61f0bcd3f17540b3461cd74e37d693f6e30a99c8c
def20b882dd96faa3e95ef/attach: connect: connection refused
```

FIGURE 12.13 – Lancement de notre image

Pour pallier à ce problème la communauté Podman nous conseille de désactiver SELinux (Figure 1.4).

```
[root@localhost ~]# setenforce 0
setenforce: SELinux is disabled
[root@localhost ~]# podman run -it --rm mageia:6
error attaching to container 79f545971332d1102dffdfc194c26ec58a98e9fdd187b52c2f27
396cda31a545: failed to connect to container's attach socket: /var/run/libpod/socket/79f545971332d1102dffdfc194c26ec58a98e9fdd187b52c2f27396cda31a545/attach: dial
unixpacket /var/run/libpod/socket/79f545971332d1102dffdfc194c26ec58a98e9fdd187b5
2c2f27396cda31a545/attach: connect: connection refused
```

FIGURE 12.14 – Lancement d'un conteneur sans SELinux

12.2.3 Conclusion

Conclusion			
	Les plus	Les Moins	Adapté ?
Podman	Léger Super sécurisé	Sous mageia 6, il n'est pas stable. il ne fonctionne pas	Non
Docker	Multiplateforme Communauté Faciliter d'utilisation	Sécurisation est plus légère que Podman Projet propriétaire	Oui
Lxc	Natif à Linux Non disponible sur Mageia	Prise en main utilisation plus complexe	Non

12.3 MMC

12.3.1 Test d'installation avec la documentation MMC

Environnement de test

Les tests pour l'installation de la MMC sont réalisés sous Mageia 6, Avec la documentation de la “Mandriva Management Console 3.11”. Test effectuer en novembre 2018

Phase de test

Réalisation des étapes indiquées dans la documentation, installation des paquets requis

```
dnf install mmc-agent mmc-web-base python-mmc-base
```

12.3. MMC

CHAPITRE 12. LES TESTS RÉALISÉS

```
● mmc-agent.service - Mandriva Management Console
  Loaded: loaded (/usr/lib/systemd/system/mmc-agent.service; enabled; vendor preset: enabled)
  Active: activating (auto-restart) (Result: exit-code) since mer. 2018-11-21 17:51:14 CET; 880ms ago
    Process: 12839 ExecStart=/usr/sbin/mmc-agent (code=exited, status=4)

nov. 21 17:51:14 localhost systemd[1]: Failed to start Mandriva Management Console.
nov. 21 17:51:14 localhost systemd[1]: mmc-agent.service: Unit entered failed state.
nov. 21 17:51:14 localhost systemd[1]: mmc-agent.service: Failed with result 'exit-code'.
```

FIGURE 12.15 – Erreur lors du lancement de l'agent de la MMC

```
23 14:25:15 localhost mmc-agent[20288]: A module named mmc-agent does not have a working installation of the service_identity module: No module named service_identity. Please install it from <https://pypi.python.org/pypi/service_identity>
23 14:25:16 localhost mmc-agent[20288]: mmc-agent.service: Control process exited, code=exited status=1
23 14:25:16 localhost mmc-agent[20288]:           Failed to start Hadoop Filesystem Management Console.
Defined-By: system
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a échoué avec le résultat Failed.
nohup: 23 14:25:16 localhost mmc-agent[20288]: mmc-agent.service Unit entered failed state.
nohup: 23 14:25:16 localhost mmc-agent[20288]: mmc-agent.service Failed with result 'exit-code'.
nohup: 23 14:25:17 localhost mmc-agent[20288]: mmc-agent.service Service hold-off time over, scheduling restart.
nohup: 23 14:25:17 localhost mmc-agent[20288]: Stopped Hadoop Filesystem Management Console.
Subject: L'unauté (unit) mmc-agent.service a terminé son arrêt
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a terminé son arrêt.
nohup: 23 14:25:17 localhost mmc-agent[20288]: Stopped Hadoop Filesystem Management Console...
Subject: L'unauté (unit) mmc-agent.service a commencé à démarer
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a commencé à démarer.
nohup: 23 14:25:18 localhost mmc-agent[20288]: A module named mmc-agent does not have a working installation of the service_identity module: No module named service_identity. Please install it from <https://pypi.python.org/pypi/service_identity>
23 14:25:18 localhost mmc-agent[20288]: mmc-agent.service Control process exited, code=exited status=1
23 14:25:18 localhost mmc-agent[20288]:           Failed to start Hadoop Filesystem Management Console.
23 14:25:18 localhost mmc-agent[20288]:           Stopped Hadoop Filesystem Management Console.
Subject: L'unauté (unit) mmc-agent.service a démarré
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a démarré avec le résultat Failed.
nohup: 23 14:25:18 localhost mmc-agent[20288]: mmc-agent.service Unit entered failed state.
nohup: 23 14:25:18 localhost mmc-agent[20288]: mmc-agent.service Failed with result 'exit-code'.
nohup: 23 14:25:19 localhost mmc-agent[20288]: mmc-agent.service Service hold-off time over, scheduling restart.
nohup: 23 14:25:19 localhost mmc-agent[20288]: Stopped Hadoop Filesystem Management Console.
nohup: 23 14:25:19 localhost mmc-agent[20288]: Starting Hadoop Filesystem Management Console...
Defined-By: system
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a terminé son arrêt.
nohup: 23 14:25:20 localhost mmc-agent[20288]: mmc-agent.service Start request repeated too quickly.
nohup: 23 14:25:20 localhost mmc-agent[20288]:           Failed to start Hadoop Filesystem Management Console.
Subject: L'unauté (unit) mmc-agent.service a échoué
Support: http://lists.freedesktop.org/mailman/listinfo/system-devel
L'unité (unit) mmc-agent.service a échoué avec le résultat Failed.
nohup: 23 14:25:20 localhost mmc-agent[20288]: mmc-agent.service Unit entered failed state.
nohup: 23 14:25:20 localhost mmc-agent[20288]: mmc-agent.service Failed with result 'start-limit-hit'.
```

FIGURE 12.16 – Journal du service de l'agent de la MMC

Il faut ensuite lancer apache pour pouvoir se connecter en local au site de la MMC

```
systemctl start httpd
```

et ensuite lancer l'agent de la MMC pour voir s'il fonctionne

```
systemctl start mmc-agent
```

Une erreur survient alors :

Figure 10.15 et Figure 10.16

L'erreur demande d'installer une dépendance python supplémentaire :

```
dnf install python-service-identity
```

Une fois installée une nouvelle erreur lors du lancement du service apparaît à nouveau :
Figure 10.17

Tentative d'installation de l'annuaire pour savoir si cela pourrait résoudre le problème. Cependant l'annuaire de la MMC est basé sur OpenLDAP, or celle-ci n'a plus les bons paquets (`openldap-mandriva-dit`) il existe uniquement un substitut (`openldap-mmcdit`) :

```
dnf install openldap-mmc-dit
```

```
● mmc-agent.service - Mandriva Management Console
   Loaded: loaded (/usr/lib/systemd/system/mmc-agent.service; enabled; vendor preset: enabled)
   Active: activating (auto-restart) (Result: exit-code) since mer. 2018-11-21 17:51:14 CET; 880ms ago
     Process: 12839 ExecStart=/usr/sbin/mmc-agent (code=exited, status=4)

nov. 21 17:51:14 localhost systemd[1]: Failed to start Mandriva Management Console.
nov. 21 17:51:14 localhost systemd[1]: mmc-agent.service: Unit entered failed state.
nov. 21 17:51:14 localhost systemd[1]: mmc-agent.service: Failed with result 'exit-code'.
```

FIGURE 12.17 – Nouvelle erreur dans le journal de l'agent

```
[root@localhost user]# /usr/share/openldap/scripts/mmc-dit-setup.sh
Please enter your DNS domain name [example.com]:
mandriva.comm

Administrator account

The administrator account for this directory is
uid=LDAP Admin,ou=System Accounts,dc=mandriva,dc=comm

Please choose a password for this account:
New password:
Re-enter new password:
{SSHA}11Z2ZqAdelcDID3Y3SaCqxmRk9guFCq0

Summary
=====
Domain:      mandriva.comm
LDAP suffix: dc=mandriva,dc=comm
Administrator: uid=LDAP Admin,ou=System Accounts,dc=mandriva,dc=comm

Confirm? (Y/n)
y
cat: /usr/share/openldap/mandriva-dit/mandriva-dit-slapd-template.conf: Aucun fichier ou dossier de ce type
Module path is empty, we hope all overlays are built-in
dirname: opérande manquant
Saisissez « dirname --help » pour plus d'informations.
dirname: opérande manquant
Saisissez « dirname -help » pour plus d'informations.
cat: /usr/share/openldap/mandriva-dit/mandriva-dit-access-template.conf: Aucun fichier ou dossier de ce type
config file testing succeeded
cat: /usr/share/openldap/mandriva-dit/mandriva-dit-base-template.ldif: Aucun fichier ou dossier de ce type
Available database(s) do not allow slapadd
ERROR
Database loading failed during test run.
Ldif file used: /tmp/mandriva-dit.bQ6k3BC5cNzv
slapd.conf file used: /tmp/mandriva-dit.ym0hg00jfUxI

Exiting
```

FIGURE 12.18 – Lancement du script de l’annuaire LDAP

Après avoir installé, il faut lancer le script d’installation pour pouvoir configurer l’annuaire avec :

/usr/share/openldap/scripts/mmc-dit-setup.sh

cependant la configuration ne fonctionne pas du tout à cause du script mal paramétré (Paramétré pour Mandriva au lieu de Mageia) : *Figure 10.18*

En essayant tout de même de lancer le site de la MMC, on constate que cela ne peut pas fonctionner sans le “mmc-agent” qui ne se lance actuellement pas : *Figure 10.19*

Conclusion

L’installation de la MMC à l’aide de la documentation n’est pas possible, car la documentation n’est pas du tout à jour. Après avoir cherché sur internet, nous pouvons trouver une documentation étant en 3.14, cependant elle est, elle-même trop datée et ne fonctionne pas sur Mageia 6.

12.3.2 Test d’installation avec les fichiers MMC ABLogix

Environnement de test

Les tests pour l’installation de la MMC sont faits sous Mageia 6, avec les fichiers que nous avions à disposition de la MMC de ABLogix et la VM utilisant cette MMC. Test

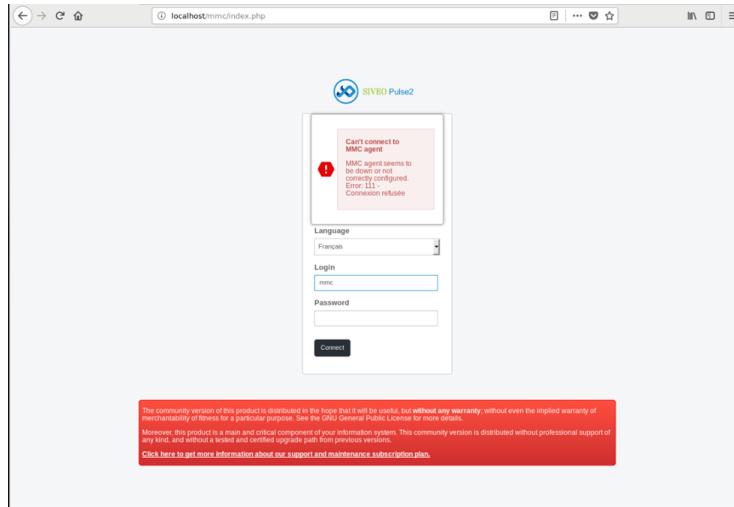


FIGURE 12.19 – Erreur lors de la connexion à la MMC

effectué en novembre 2018.

Phase de test

Récupération des fichiers de la MMC. Comme aucun installateur est présent, mais uniquement tous les fichiers dans le dossier test de copie de tous les fichiers dans les bons répertoires. Le résultat se révèle sans succès, car il est impossible de lancer le service "mmc-agent" encore une fois. Le test de la VM mis à notre disposition est également un échec, car elle est corrompue et donc ne fonctionne pas.

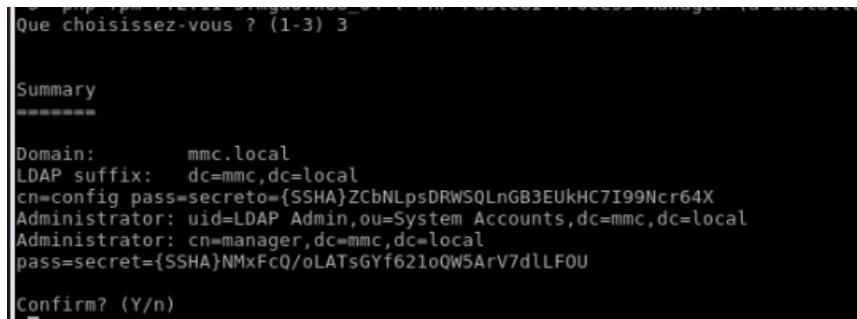
Conclusion

L'installation de la MMC à l'aide des fichiers est elle aussi un échec, car aucun service ne se lance et il n'y a pas de possibilités de la faire fonctionner sans modifier de nombreux fichiers, ce qui peut consommer énormément de temps de prises en main de la MMC.

12.3.3 Test d'installation avec un docker

Environnement de test

Les tests pour l'installation d'une MMC préconfigurée sur docker, essayée sous Mageia 6 Test effectués en novembre 2018.



```
Que choisissez-vous ? (1-3) 3

Summary
-----
Domain: mmc.local
LDAP suffix: dc=mmc,dc=local
cn=config pass=secreto={SSHA}ZCbNLpsDRWSQLnGB3EUkHC7I99Ncr64X
Administrator: uid=LDAP Admin,ou=System Accounts,dc=mmc,dc=local
Administrator: cn=manager,dc=mmc,dc=local
pass=secret={SSHA}NMxFcQ/oLATsGYf62loQW5ArV7dLLFOU

Confirm? (Y/n)
```

FIGURE 12.20 – Erreur lors du lancement du conteneur contenant une MMC

Phase de test

Récupération du conteneur :

```
docker pull osixia/mmc-agent
```

Lancement du conteneur :

```
docker run mmc-agent
```

Une erreur survient lors du lancement du conteneur de la MMC : *Figure 10.20*

Conclusion

Le conteneur ne fonctionnant pas, cette solution n'est pas utilisable. Après des tests d'autre docker, je n'ai pas trouvé de docker avec une MMC préconfigurer pour le moment.

12.3.4 Solution potentielle

Après quelques tests de modification de fichier pour voir s'il était possible de régler quelques problèmes lors des installations. Les solutions pour avoir une MMC utilisable rapidement pour notre projet ne sont pas nombreuses et coûteuses en temps. La solution qui est la moins couteuse en temps pour le moment est de trouver une application similaire à la MMC au niveau des fonctionnalités. Cependant nous avons demandé au mainteneur des paquets de les mettre à jour ce qui résoudra normalement nos soucis. De plus il nous offre son aide pour l'utilisation de la MMC ce qui permettra d'avancer plus vite dans la réalisation de la MMC.

12.4 FirewallD

Dans ce tutoriel nous utilisons Mageia.

```
[root@localhost user]# firewall-cmd --zone=drop --list-all
drop (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

FIGURE 12.21 – list-all_drop

```
anthony@pc-anthony:~$ ping 10.105.201.198
PING 10.105.201.198 (10.105.201.198) 56(84) bytes of data.
^C
--- 10.105.201.198 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10232ms
```

FIGURE 12.22 – ping1

12.4.1 Nos premiers tests

Pour les tests, nous sommes partis d'une machine virtuelle utilisant l'OS Mageia sur une machine hôte utilisant Ubuntu. Le paramétrage de Firewalld se fera sur la machine virtuelle. Comme dit dans la section zones, nous avons décidé de partir de la zone drop qui bloque toutes les communications.

Test avec la commande ping

Ajoutons la carte réseau de la machine virtuelle dans la zone drop :

```
# firewall-cmd --permanent --zone=drop --add-interface=enp0s3
```

On recharge Firewalld :

```
# firewall-cmd --reload
```

Et on affiche les informations de la zone drop (cf figure list-all_drop). Maintenant testons de pinger la machine virtuelle avec notre hôte (cf figure ping1). On remarque que le ping est rejeté avec aucun message de retour comme prévu avec la zone drop.

Ajoutons maintenant une règle autorisant notre hôte à communiquer avec la machine virtuelle :

```
# firewall-cmd --permanent --zone=drop --add-rich-rule='rule
family=ipv4 source address=<@ip_Hote> accept'
```

On recharge firewalld :

```
anthony@pc-anthony:~$ ping 10.105.201.198
PING 10.105.201.198 (10.105.201.198) 56(84) bytes of data.
64 bytes from 10.105.201.198: icmp_seq=1 ttl=64 time=0.547 ms
64 bytes from 10.105.201.198: icmp_seq=2 ttl=64 time=0.426 ms
64 bytes from 10.105.201.198: icmp_seq=3 ttl=64 time=0.577 ms
64 bytes from 10.105.201.198: icmp_seq=4 ttl=64 time=0.610 ms
^C
--- 10.105.201.198 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3077ms
rtt min/avg/max/mdev = 0.426/0.540/0.610/0.069 ms
```

FIGURE 12.23 – ping2

```
[root@localhost user]# ping 10.105.201.198
PING 10.105.201.198 (10.105.201.198) 56(84) bytes of data.
^C
--- 10.105.201.198 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6155ms
```

FIGURE 12.24 – ping3

```
# firewall-cmd --reload
```

Et on reteste le ping (cf figure ping2). Cette fois on remarque bien que le ping arrive à destination. Testons maintenant à partir d'une autre machine (cf figure ping3). Celle-ci ne peut pas communiquer. Notre règle a donc bien fonctionné.

Test d'une connexion SSH

L'objectif maintenant est de n'autoriser que les connexions SSH venant de notre machine hôte.

Supprimons l'ancienne règle :

```
# firewall-cmd --permanent --zone=drop --remove-rich-rule=
rule family=ipv4 source address=<@ip_hote> accept'
```

Ajoutons la nouvelle :

```
# firewall-cmd --permanent --zone=drop --add-rich-rule='rule
family=ipv4 source address=<@ip_hote> service name=ssh
accept'
```

Rechargeons Firewalld et vérifions la présence de la nouvelle règle (cf figure list_all_ssh). La règle a bien été modifiée. Il ne reste plus qu'à tester le ping et la connexion SSH.

```
[root@localhost ~]# firewall-cmd --list-all --zone=drop
drop (active)
  target: DROP
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
    rule family="ipv4" source address="10.105.201.79" service name="ssh" accept
[root@localhost ~]#
```

FIGURE 12.25 – list_all_ssh

```
anthony@pc-anthony:~$ ping 10.105.201.198
PING 10.105.201.198 (10.105.201.198) 56(84) bytes of data.
^C
--- 10.105.201.198 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3075ms
```

FIGURE 12.26 – ping4

Pour le ping on remarque que la communication ne marche pas comme convenu (cf figure ping4). Pour la connexion ssh on remarque qu'elle fonctionne (cf figure ssh1).

Test d'une connexion via le port 22

Maintenant nous allons tester d'autoriser les connexions uniquement via le port 22 qui sert au service SSH. On supprime l'ancienne règle et on ajoute la nouvelle :

```
# firewall-cmd --permanent --zone=drop --remove-rich-rule='
rule family=ipv4 source address=<@ip_hote> service name=ssh
accept'
# firewall-cmd --permanent --zone=drop --add-rich-rule='rule
family=ipv4 source address=<@ip_hote> port port=22 protocol
=tcp accept'
```

On recharge ensuite firewalld.

```
anthony@pc-anthony:~$ ssh user@10.105.201.198
Password:
Last login: Tue Nov 13 10:55:57 2018 from 10.105.201.79
[user@localhost ~]$
```

FIGURE 12.27 – ssh1

```

anthony@pc-anthony:~$ ping 10.105.201.198
PING 10.105.201.198 (10.105.201.198) 56(84) bytes of data.
^C
--- 10.105.201.198 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2049ms

anthony@pc-anthony:~$ ssh user@10.105.201.198
Password:
Last login: Tue Nov 13 10:56:10 2018 from 10.105.201.79
[user@localhost ~]$ █

```

FIGURE 12.28 – ping+ssh

```
# firewall-cmd --reload
```

Maintenant testons le ping et la connexion SSH. On remarque encore une fois que le ping ne marche pas mais que la connexion SSH fonctionne toujours (cf figure ping+ssh). Notre règle a donc marché.

Autorisation de toutes les communications via un service ou un port

Il est possible d'autoriser toutes les communications via un service ou un port en l'ajoutant à la zone :

```

# firewall-cmd --permanent --zone=<zone choisie> --add-service
  =<service a ajouter>
# firewall-cmd --reload

# firewall-cmd --permanent --zone=<zone choisie> --add-port=<
  port a ajouter>
# firewall-cmd --reload

```

12.4.2 Port forwarding

Ici nous souhaitons faire une redirection de port pour permettre la connexion entre deux machines qui ne sont pas sur le même réseau local. Dans notre exemple, nous avons 3 machines (cf schéma port forwarding).

Dans l'exemple nous allons utiliser le service SSH, nous allons rediriger tout ce qui vient du port 22 du pc1 et qui arrive sur le port 22 de la carte enp0s3 de la vm FirewallD vers le port 22 du pc2 (voir flèche du schéma).

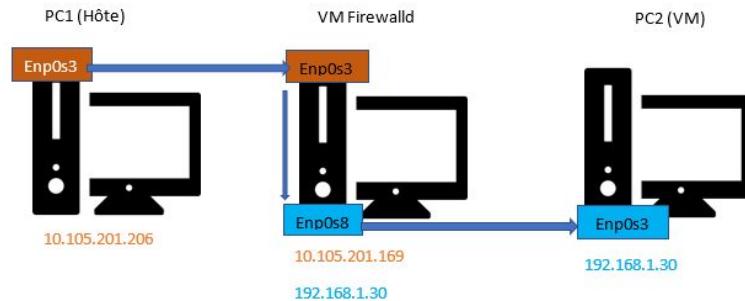


FIGURE 12.29 – schéma port-forwarding

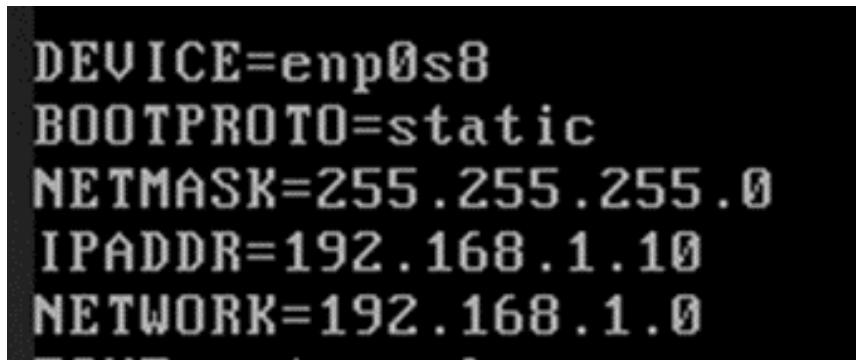


FIGURE 12.30 – image ifcfg

Attribution des IPs

Avant d'attribuer les ips, il faut ajouter une carte réseau à la VM Firewalld puis lui affecter une IP statique (192.168.1.10). Il faut également mettre une IP statique (192.168.1.30) au pc2. Pour cela, sur Mageia, il faut aller dans /etc/sysconfig/network-script/ifcfg-nom_de_la_carte et écrire (cf image ifcfg).

Paramétrage de Firewalld

On ajoute les deux cartes réseau dans la zone drop :

```
# firewall-cmd --permanent --zone=drop --add-interface=enp0s3
# firewall-cmd --permanent --zone=drop --add-interface=enp0s8
```

Il faut activer le masquerade :

```
# firewall-cmd --permanent --zone=drop --add-masquerade
```

Ajoutons maintenant à la zone drop une règle autorisant le PC1 à communiquer via le port 22 :

```
# firewall-cmd --permanent --zone=drop --add-rich-rule='rule
family=ipv4 source address=10.105.201.206 port port=22
protocol=tcp accept'
```

Et enfin le port forwarding (on redirige tout ce qui arrive sur le port 22 de la machine Firewalld vers le port 22 du PC2) :

```
# firewall-cmd --permanent --zone=drop --add-forward-port=
port22:proto=tcp:toport=22toaddr=192.168.1.30
```

Chapitre 13

Presentation du projet

Le jeudi 28/03/2019 nous avons présenté notre projet lors de l'open esiea. Le projet n'étant pas très visuel nous avons décidé de créer une maquette (cf Maquette) pour faciliter la compréhension du jury. Pour confectionner cette dernière, nous avons utilisé l'imprimante 3D de l'école ainsi que des playmobsils. Les playmobsils nous ont été prêtés par Clémence Peslerbes et Paul Letourneau.

En plus de notre maquette, nous avons effectué une démonstration de notre travail réalisé (cf schemaPresentation).

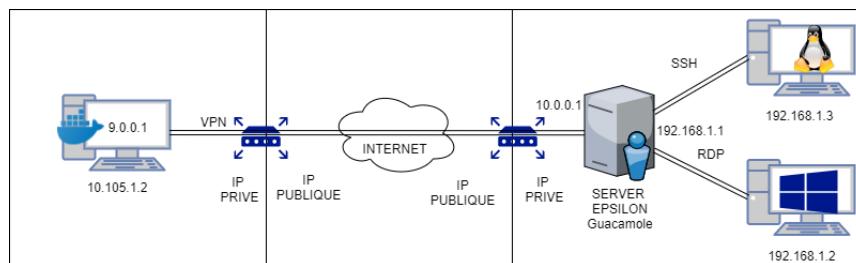


FIGURE 13.1 – schemaPresentation

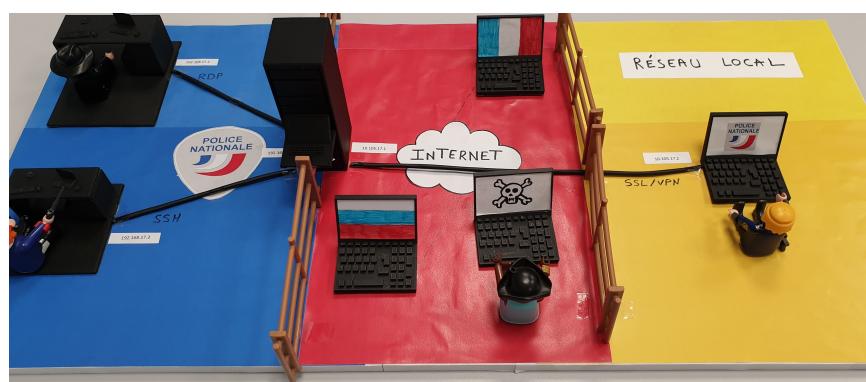


FIGURE 13.2 – Maquette

Chapitre 14

Compte-rendu de réunion 1 - 9

On trouvera à la suite l'ensemble des rapports de réunion que nous avons eu avec nos suiveurs ainsi que les réunions que nous avons réalisées avec Alexandre Dey. Alexandre a réalisé un premier POC durant sa période de stage de 4ème année.

Compte rendu n°1



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Anthony BILLETTE
DATE DE LA REUNION	10/09/2018
HEURE	14:30 à 16:00
LIEU	ESIEA Ouest 38 Rue des Docteurs Calmette et Guérin 53000 Laval
TYPE DE REUNION	Lancement du PST
REDACTEURS	Anthony BILLETTE, Anthony YAR, Jean-Baptiste PESLERBES
PARTICIPANTS	Jean-Pierre AUBIN, Jean-François BELLANGER et ses adjoints. Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

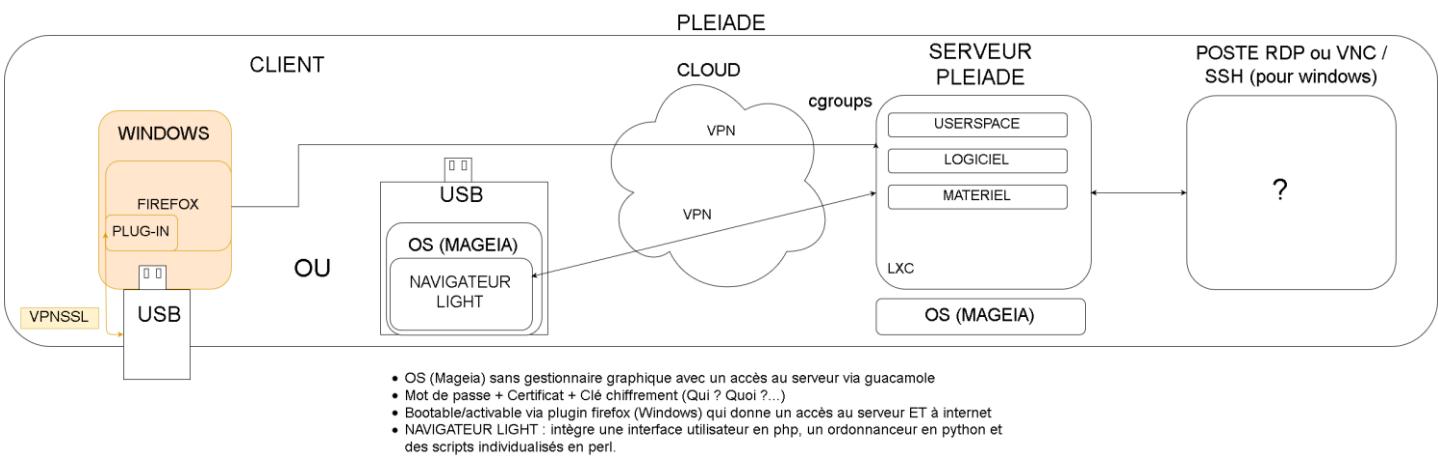
Rubriques à l'ordre du jour

DISCUSSION	REUNION DE LANCEMENT DU PST
	Présentation des différents membres du projet. Définition du projet ainsi que son contexte d'utilisation, explication du travail déjà réalisé, prédéfinition des objectifs à réaliser durant l'année.
	Nous avons décidé de nous consacrer sur le projet « EPSILON ». En fonction de l'avancement de ce dernier, nous nous tournerons sur le projet « IGOINE »
	Création d'un schéma explicatif du projet et de l'utilité de chaque module. Celui-ci sera placé dans notre redéfinition du projet.
	Les objectifs par ordre chronologique :
	<ul style="list-style-type: none">- Comprendre et tester « Pléiade ».- Définir les objectifs à atteindre dans l'année.- Voir comment fonctionne conteneurBox.- Démarrer un OS avec un navigateur à la place d'un environnement graphique.- L'OS doit intégrer guacamole pour se connecter au serveur. (https://guacamole.apache.org/)- Mise en place de l'OS personnalisé sur une clé bootable.- Test du lancement de l'OS à partir de la clé USB.- Création du plug-in sur Firefox sans démarrage de L'OS présent sur la clé USB.



DISCUSSION	ORGANISATION DU TRAVAIL
	Les rapports seront à envoyer à Jean-Pierre Aubin et à Jean-François Bellanger et ses adjoints à l'adresse (ddsXXX@interieur.gouv.fr)
DISCUSSION	AVANT LA PROCHAINE REUNION
	<ul style="list-style-type: none"> - Réécriture du dossier de description du projet. - Rédaction de l'état de l'art. - Etablir des objectifs dans le but de cadrer le projet. - Contacter Alexandre Dey. - Réfléchir à de futurs axes d'amélioration. - Organisation de la future réunion.

DISCUSSION	PROCHAINE REUNION	
	Les prochaines réunions seront réalisées les jeudis après-midi dans la mesure du possible. Elles auront lieu toutes les 2-3 semaines. La prochaine réunion aura lieu au poste de police.	
PROCHAINE REUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER et ses adjoints Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police.	A définir aux alentours de fin septembre



Compte rendu n°2



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Théo PORTIER
DATE DE LA REUNION	10/09/2018
HEURE	18:30 à 22:00
LIEU	RENNES
TYPE DE REUNION	Compréhension du sujet (réunion annexe)
REDACTEURS	Anthony BILLETTE, Simon RUFFET, Jean-Baptiste PESLERBES
PARTICIPANTS	Alexandre DEY, Marie KERGUELEN, Quentin JEANNAUD.
	Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	REUNION ANNEXE
	Après une réunion avec Alexandre DEY (ancien développeur du projet), il nous a conseillé de ne pas installer PLEIADE mais plutôt de s'en inspirer. Pour commencer il pense qu'installer un serveur avec juste un VPN suffira du coté serveur. Pour le coté client, on commencera avec l'installation du VPN sur un OS light.
	Pour le VPN, il nous conseil de regarder Wireguard car c'est un VPN intégré à linux et plus sécurisé par sa conception car il n'utilise qu'un seul protocole. Ceci évite les attaques qui consistent à forcer l'utilisation du protocole le plus faible.
	Pour la conteneurisation, il nous a préconisé d'utiliser LXC/LXD par rapport à Docker et Kata Containers car c'est le seul qui a été pensé pour la conteneurisation d'OS dès sa conception.
	Il nous a conseillé d'utiliser containerBox pour gérer les différents conteneurs. Ce dernier est basé sur LXD et il permet de réaliser des conteneurs facilement. Pour les autres conteneurs on utilisera LXD/LXC.
	Pour l'OS, il nous a proposé de regarder ALPINE car c'est un OS léger cependant il faut que l'on regarde s'il est compatible avec LXC/LXD.

Pour la partie organisation, il nous a conseillé de diviser le projet en trois parties :

- Création d'une clé bootable avec un OS light et démarrage du navigateur sans autre interface graphique.
- Monter un VPN entre deux machines avec le VPN que nous choisirons.
- Gérer et créer la communication entre le PC serveur et le PC client. Le but étant de connecter l'interface web avec le serveur pléiade. Le serveur pléiade donne un accès au logiciel guacamole qui se connecte à une machine virtuelle.

Pour terminer, il nous a vivement recommandé d'avoir deux ordinateurs de travail fixe pour pouvoir travailler sans risque pour nos données personnelles ainsi qu'une salle de travail nous permettant de nous réunir avec notre matériel. Le fait de travailler avec la méthode Agile nous oblige à des réunions très régulières et par conséquent un local de travail devient très utile.

Compte rendu n°3



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Anthony BILLETTE
DATE DE LA REUNION	21/09/2018
HEURE	14:45 à 17:15
LIEU	DDSP LAVAL
TYPE DE REUNION	Compréhension précise du projet
REDACTEURS	Anthony BILLETTE, Anthony YAR
PARTICIPANTS	Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	MATERIELS
Nous avons rendu l'excédent de matériel que nous avions (claviers, souris, câbles). Nous avons récupéré par la même occasion un écran Samsung, ainsi qu'une clé USB de 16GO.	

Remise de ressource documentaire.

DISCUSSION	POINTS ABORDÉS DURANT LA REUNION
- Présentation de notre état de l'art. - Présentation de notre schéma représentant l'ensemble du projet. - Présentation du schéma représentant l'ensemble du projet créé par Jean-François. - Explication de la rupture protocolaire. - Présentation de 4 projets ressemblant au nôtre. (NEO pour les systèmes embarqués, Messagerie Nomade 2, connexion à distance avec SPAN, CLIP OS avec un système de couche basse, couche haute) ainsi que le périphérique de confiance. - Réponses à nos questions :	Les cgroups permettent de gérer les droits et les règles entre les conteneurs, ils sont la principale sécurité dans le cloisonnement des conteneurs. Pourquoi utiliser des conteneurs ? Ils nous servent de sécurité puisqu'ils permettent un isolement avec la machine hôte. Explication des ruptures protocolaires exemples :

- Navigateur -> serveur Pleiade (https)
 - Serveur Pleiade -> serveur distant (RDP)
 - Serveur distant -> Mail (IMP)
 - Serveur distant -> Fichier (SMB)
- C'est la technique de Vauban.

Freerdp // guacamole : deux outils identiques pour la connexion à une machine hôte à distance.

DISCUSSION

A VOIR AVEC M.REY

Soit partir de l'OS d'alcasar en ajoutant ce dont nous avons besoin, ou partir d'un OS Mageia et enlever ce qui ne nous sert pas.

Le problème d'alcasar est qu'il ne contient pas LXC et nous ne savons pas s'il est possible de l'ajouter.

Voir pour l'erreur de Display qui apparaît au lancement de Firefox.

DISCUSSION

POUR LA PROCHAINE REUNION

La prochaine réunion sera réalisée les jeudis après-midi dans la mesure du possible. Elle aura lieu dans un délai de 2 à 3 semaines.

- Réaliser un état des lieux de l'avancement du projet
- Début de la rédaction du dossier en adéquation avec l'avancement.

PROCHAINE REUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER et ses adjoints Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police / ESIEA	A définir aux alentours de mi-octobre

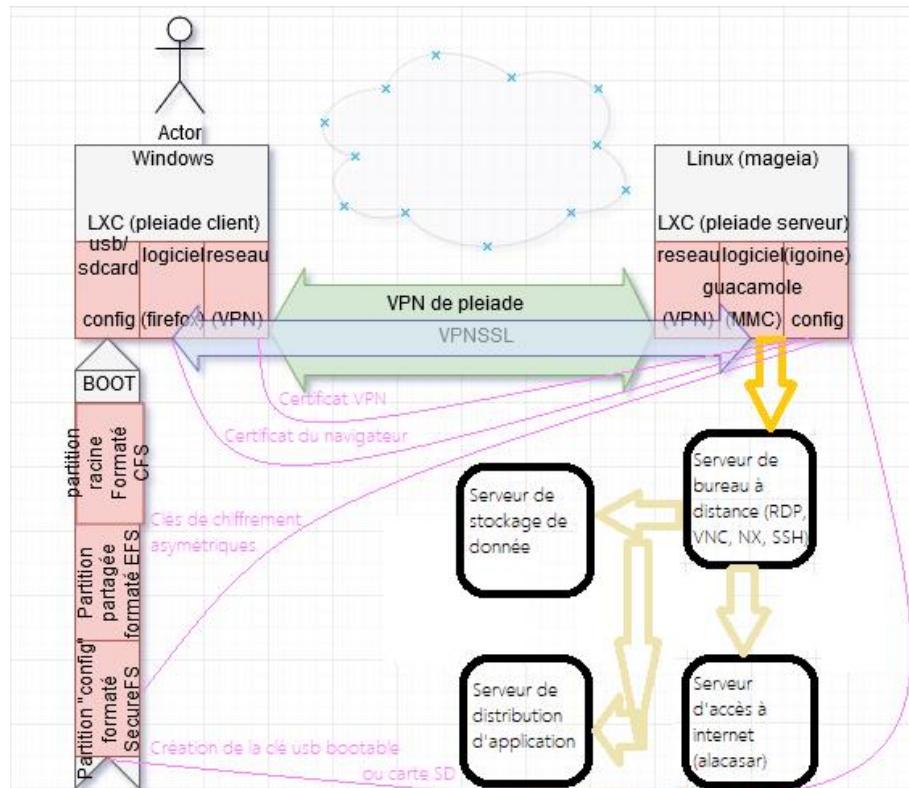


Schéma du projet réalisé par Jean-François

Compte rendu n°4



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Jean-Baptiste Peslerbes
DATE DE LA REUNION	11/10/2018
HEURE	18h30 à 20h00
LIEU	Rennes
TYPE DE REUNION	Installation de PLEIADE
REDACTEURS	Simon Ruffet, Jean-Baptiste Peslerbes
PARTICIPANTS	Alexandre DEY Simon Ruffet, Jean-Baptiste Peslerbes

Rubriques à l'ordre du jour

DISCUSSION	INSTALLATION DE PLEIADE
	<p>Le but de cette réunion était d'installer PLEIADE server/client sur des machines virtuelles avec l'aide d'Alexandre DEY (fondateur du projet). Nous avons découvert que le projet n'était pas opérationnel. En effet, il s'agit actuellement d'un POC. Il manque au projet de la documentation et des scripts d'installation et de lancement de pléiaide. De plus, certains bugs n'ont pas été résolus pendant la création de ce POC.</p> <p>Il en résulte que nous ne pouvons pas créer le module complémentaire EPSILON sachant que PLEIADE n'est pas fonctionnel.</p>
DISCUSSION	SOLUTIONS ENVISAGABLES
	<p>La finalisation de PLEIADE va être obligatoire pour pouvoir continuer le module EPSILON. C'est pour cela que nous proposons de faire une refonte des objectifs pour la fin du premier semestre. Le premier livrable serait donc un PLEIADE fonctionnel, automatisé et documenté.</p> <p>Ensuite, nous pourrons réaliser un deuxième livrable qui sera notre module PLEIADE s'il nous reste suffisamment de temps pour cela.</p>

DISCUSSION	FAISABILITE DU PROJET	
Suite à des échanges avec Alexandre DEY, certains points du projet s'avèrent non réalisables avec nos compétences et le temps imparti.		
PROCHAINE REUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER et ses adjoints Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Préfecture	18 Octobre 2018 à 14h30

Compte rendu n°5



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Frédéric Arrighi
DATE DE LA RÉUNION	18/10/2018
HEURE	De 14 h 30 à 18 h
LIEU	Préfecture de Laval 46 Rue Mazagran 53000 Laval
TYPE DE RÉUNION	Réunion d'avancement du projet, réaffectation des objectifs
RÉDACTEURS	Théo PORTIER, Simon RUFFET, Jean-Baptiste PESLERBES, Anthony BILLETTE, Anthony YAR
PARTICIPANTS	Jean-François BELLANGER, Frédéric Arrighi, Olivier, François, Cyril, Paul Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	INTRODUCTION/PROBLÉMATIQUE
	Le client a besoin d'accéder à deux réseaux différents. L'un sécurisé (RIE) et l'autre un peu moins (ADSL). Ces deux réseaux ne doivent pas pouvoir communiquer entre eux.
	Quatre solutions s'offrent à nous :
	1-Le client possède deux ordinateurs et transfert ses fichiers à l'aide d'une clé USB. 2-Le PC du client possède deux OS (Dual boot) et doit redémarrer pour faire son échange de fichier via USB. 3-On virtualise un OS (VM) et on lui offre un réseau dédié. L'hôte devra être plus sécurisé que sa VM. Les deux seront étanches l'un par rapport à l'autre. 4-PLEIADE
	La solution retenue est PLEIADE car elle semble la plus optimisée.

DISCUSSION	DES POINTS A REVOIR DANS PLEIADE
Voici les différents points qui ne vont pas pour la version actuelle de PLEIADE :	
	<ul style="list-style-type: none"> - Il n'y a pas de version de PLEIADE, nous la définissons donc à la version : V0.1 - Il n'y a pas d'état de développement, nous le définissons donc à alpha. - Il n'y a pas d'installateur fonctionnel. - Il n'y a pas de paquetage pour LXD, containerbox, PLEIADE, EPSILON. - Les dépendances ne sont pas précisées - Il n'y a pas de documentation - Le code n'est pas prévu pour fonctionner sur Mageia - Alexandre n'a pas utilisé d'atelier de génie logiciel (git, subversion, vagrant)
DISCUSSION	POINTS TECHNIQUE (À TESTER)
Tout d'abord, Frédéric Arrighi nous a fait un POT (Proof Of Technology) sur l'utilisation d'un bash de la couche Linux présente sur Windows (via Ubuntu). Il nous a fait une démonstration en lançant firefox (Linux) à partir du bash sur le serveur d'affichage Xming.	
	Sachant que cela est possible, il est éventuellement réalisable d'installer des conteneurs LXC sur la couche Windows. Ainsi, nous pourrons lancer PLEIADE sur un Windows. Pour être certain que cela puisse fonctionner de manière sécurisée, il faut tester les points suivants :
	<ul style="list-style-type: none"> • Tester Xming et faire un état de l'art sur l'existence éventuelle d'un serveur x plus sécurisé/Legé. • Concernant la solution EPSILON Windows, la POT de Frédéric nous a montré qu'il était possible d'utiliser un Ubuntu sur un Windows sans virtualisation. Il faut maintenant vérifier si cet environnement est viable et s'il est possible d'installer des conteneurs LXC sur cet Ubuntu.
Nous rencontrons également d'autres soucis :	
	<p>Il faut tester s'il est possible de faire fonctionner les conteneurs LXC lancés par des utilisateurs sur MAGEIA. Pour le moment, nous réussissons à créer des conteneurs privilégiés et non privilégiés, mais seulement à l'aide du root. Cela pose un problème, car les postes sur lesquels nous devons déployer les conteneurs ne possèdent pas les droits administrateur. Cette fonctionnalité est pourtant offerte par LXC. Cependant, il n'existe pas suffisamment de documentation sur MAGEIA pour que nous puissions le faire.</p>
	<p>Une des solutions possibles données par Monsieur Bellanger serait d'installer le paquet Docker sur MAGEIA. Ainsi, ce contenu nous donnera accès à cette fonctionnalité. À tester. Si cela fonctionne, on pourra garder MAGEIA comme OS. Sinon il faudra trouver une autre alternative qui soit la plus semblable possible à MAGEIA, les policiers étant formés uniquement sur MAGEIA.</p>
	<p>Ensuite, un problème concernant les templates de conteneurs offerts par LXC se pose. En effet, il n'existe à ce jour aucun template MAGEIA disponible. Cependant, la V6 de MAGEIA partage à présent le même gestionnaire de package que FEDORA (DNF). L'idée serait donc d'utiliser un template FEDORA, qui posséderait donc la même méthode d'installation que MAGEIA. Il faut tester si cette solution est viable (Installation semblable à MAGEIA).</p>

Nous avons des recherches à effectuer :

- Packager des logiciels
- S'il est possible de déchiffrer les trois partitions de la clé au moment du boot.
- S'il est bien ou non de remplacer le VPN par Wireguard. Vérifier que le VPN est non débrayable.
- Recherche d'information sur SecureFS pour la clé USB.

Nouvelles contraintes :

- Ne pas utiliser de Proxy
- L'installateur doit ressembler à celui d'Alcasar

DISCUSSION	ORGANISATION
Voici les différents points que nous avons abordés pendant la réunion et que nous devons décider sur notre organisation :	
<ul style="list-style-type: none">- Qui valide le code ? Qui l'intègre ? Qui le push ?- Condition des tests unitaire et intégration.- Conclusion sur les tests et remédiation	

DISCUSSION	PROCHAINE REUNION
<p>Réalisation de la documentation technique pour le projet PLEIADE (prérequis, développement, installation, utilisation).</p> <p>Mise en route de PLEIADE.</p> <p>Réalisation de l'installateur.</p> <p>Création d'un livret sur les informations que nous avons sur PLEIADE</p>	

PROCHAINE RÉUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police.	À définir aux alentours de début novembre

Compte rendu n°6



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Anthony Billette
DATE DE LA RÉUNION	08/11/2018
HEURE	De 14 h 30 à 18 h
LIEU	Poste de police
TYPE DE RÉUNION	Réunion d'amélioration du projet et des objectifs
RÉDACTEURS	Théo PORTIER, Simon RUFFET, Jean-Baptiste PESLERBES, Anthony BILLETTE, Anthony YAR
PARTICIPANTS	Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	INTRODUCTION/PROBLEMATIQUE
	L'objectif de la réunion était de présenter l'avancement de chaque membre de l'équipe (chacun travaille sur un domaine particulier du projet) pour ainsi définir quelles technologies nous allons utiliser. Nous avons également évoqué la possibilité de stage au sein de la police et voir où en était la démarche.
DISCUSSION	CONTENEURS
	Concernant les conteneurs, nous avions le choix entre Docker, PODMAN et LXC. Après quelques études, il s'avère que Docker est un LXC amélioré avec une interface graphique particulière. LXC ayant de nombreux désavantages (Ancien et limité, pas de documentation, aucun template de Mageia disponible) nous nous orientons donc vers la solution Docker. Concernant PODMAN, cette technologie est trop jeune pour être utilisée actuellement. Elle n'est pas assez stable. De plus, il n'y a pas de réponse des développeurs. PODMAN reste une solution pour l'avenir, il faut juste qu'elle soit plus aboutie. Comme technologie de conteneurisation, nous allons donc utiliser Docker en ligne de commande. Il faudra donc supprimer les scripts d'interfaces graphiques (/etc/docker). Pour la prochaine réunion, il faudra faire un document justifiant ce choix.
DISCUSSION	VPN

Concernant la technologie du VPN, nous avons pour obligation d'utiliser un VPN ayant la possibilité de faire du peer to peer (P2P). En effet, il est prévu pour les prochaines années que 2 clients puissent communiquer entre eux sans passer par le serveur. Par conséquent, tout les VPN ancien tels que OpenVPN, IPsec, ... ne peuvent pas convenir à notre projet.

Il nous reste par conséquent la possibilité d'utiliser des VPN hybrides (P2P & Client/serveur). Nous pouvons donc utiliser Wireguard et Freelan.

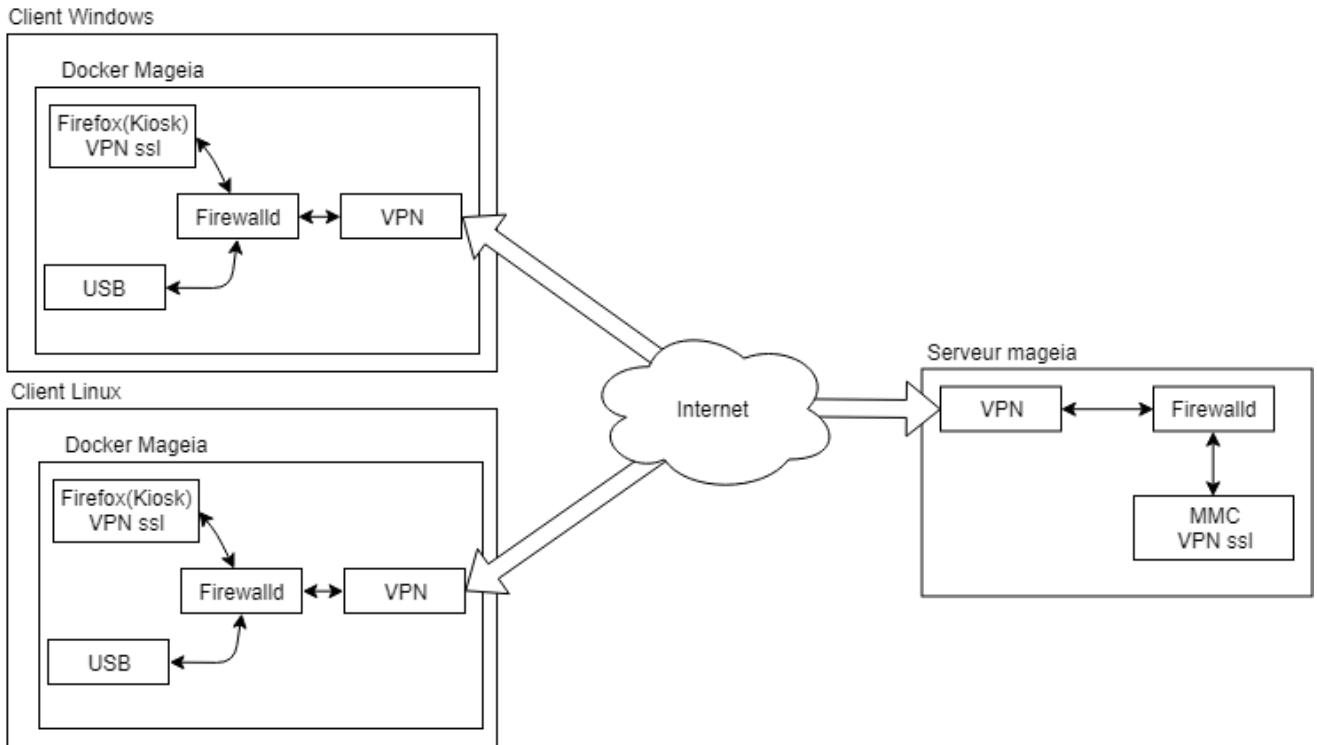
Cependant, concernant Wireguard, leur site stipule : "WireGuard n'est pas encore terminé. *Vous ne devriez pas compter sur ce code.* Il n'a pas fait l'objet d'un audit de sécurité approprié et le protocole est toujours susceptible d'être modifié"

Nous optons donc pour Freelan.

Néanmoins, cette solution reste compliquée car il n'existe pas d'installateur pour Mageia. Nous allons donc contacter les développeurs pour avoir de l'aide. A terme, nous espérons créer un package rpm.

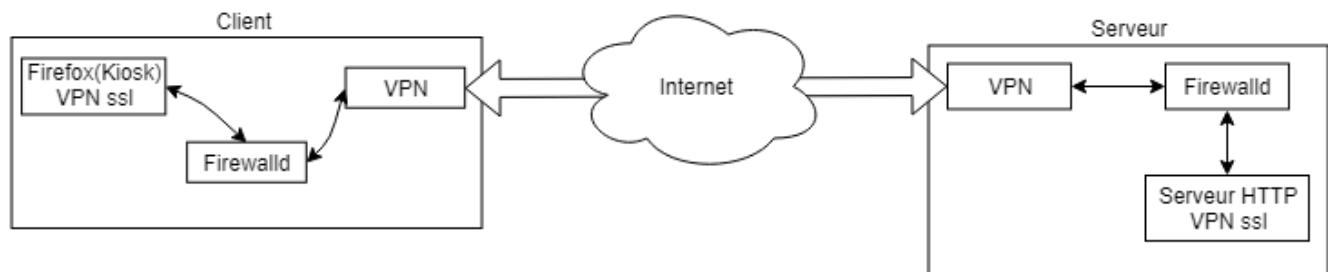
DISCUSSION	MMC
	La MMC (Mandriva Management Center) permet de réaliser de la défense en profondeur en utilisant la rupture protocolaire 3 tiers (principe de Vauban).
	Actuellement il en existe plusieurs, nous avons accès à deux d'entre-elles (Nous ne pouvons pas utiliser la MMC du ministère de l'intérieur (CHEOPS NG) car elle est protégée). La MMC de ABlogix et celle de Julien Vandeschricke fournit par notre suiveur. Pour notre projet nous allons réutiliser une des solutions déjà existantes et l'adapter à notre besoin.
	La MMC de Mr Vandeschricke n'est pas adaptée car elle utilise un module professionnel n'étant pas nécessaire pour notre projet (PULSE 2).
	Ainsi, nous choisissons celle de ABlogix puisqu'elle utilise des modules utiles pour notre projet. A présent, il faut réussir à l'installer et l'exploiter.
DISCUSSION	FIREWALL
	Pour communiquer de manière régulée entre les conteneurs, il fallait que nous instaurions des règles de communication. Plusieurs solutions s'offraient à nous.
	Dans un premier temps, nous avons commencé avec iptables. Néanmoins, nous avons rencontré un problème lors du démarrage du firewall par défaut de Mageia (Shorewall). En effet, toute la configuration des iptables étaient modifiées par Shorewall.
	Par conséquent, nous avons continué sur Shorewall. Cependant, il ne convient pas au projet puisqu'il est vieillissant et ne fonctionne pas de manière dynamique.
	C'est pourquoi nous avons continué nos recherches sur un autre firewall (Firewalld). Ce dernier répond parfaitement à nos besoins car il fonctionne dynamiquement et permet la création de règles permanentes. De plus, il a été utilisé par Alcasar donc au cas où nous pourrions demander de l'aide à Mr REY.
	Nous avons choisi pour Firewall Firewalld. Il nous reste à l'installer et à l'exploiter dans notre projet.
DISCUSSION	REDEFINITION DU PROJET

Nous avons reçu un nouveau besoin du client. Il souhaite sur linux avoir accès à un bureau pour pouvoir profiter de quelques fonctionnalités comme lire ses pdf. Ce schéma ci-dessous représente la redéfinition de notre projet :



Aux vues des nouvelles contraintes, il est optionnel de faire une clé bootable. En effet, chaque utilisateur utilise déjà un environnement (Windows ou Linux). Cela nous facilite la tâche puisque Docker nous permet de créer un conteneur MAGEIA quelque soit l'environnement. Cela signifie que nous n'avons besoin de faire qu'une seule installation propre à Mageia.

Pour premier objectif, nous voulons réaliser cette architecture :



Nous avons terminé par aborder le thème de la documentation. Nous devons donc réaliser différentes documentations pour les différents utilisateurs finaux du projet.

De plus, afin de protéger notre code nous utilisons la licence Cecill car celle-ci est adaptée à la législation française.

PROCHAINE RÉUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police.	A définir aux alentours de fin novembre

Compte rendu n°7



PST PLEIADE (EPSILON)

REUNION PAR	Anthony Billette
DATE DE LA RÉUNION	06/12/2018
HEURE	De 10 h 30 à 12 h
LIEU	Poste de police
TYPE DE RÉUNION	Présentation de l'avancement du projet
RÉDACTEURS	Théo PORTIER, Simon RUFFET, Jean-Baptiste PESLERBES, Anthony BILLETTE, Anthony YAR
PARTICIPANTS	Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	INTRODUCTION/PROBLÉMATIQUE
L'objectif de la réunion était de présenter l'avancement de chaque membre de l'équipe (chacun travaille sur un domaine particulier du projet). Nous avons également abordé les difficultés liées aux financements pour nos stages.	
DISCUSSION	AVANCEMENT
Le premier point, les avantages de la MMC sont les facilités de développement de l'outil grâce à ces plug-ins. En effet pour ajouter une fonctionnalité à la MMC il suffit de rajouter un plug-in adapté à nos besoins. La MMC est mise à jour par les développeurs de chez Mageia, cela nous diminue la maintenance. L'ajout des modules Jitsi / Qwant / thunderbolt est facilité à l'aide du module de la MMC, pulse. Le deuxième point, nous arrivons actuellement à réaliser le projet, mais notre méthode n'est pas sécurisée. Nous arrivons à réaliser la gateway dans les conteneurs. Le troisième point, nous ne pourrons pas utiliser SELinux car ce dernier est un projet initialement réalisé par la NSA. Nous ne faisons pas confiance aux entités américaines. Nous utiliserons par conséquent Msec l'outil développé par Mageia pour leur sécurité. Le quatrième point, concernant firewallID, la documentation est finie et des tests ont été réalisés.	

Le cinquième point, nous avons réussi à packager freelan, et aussi réalisé l'image docker de freelan. Nous sommes actuellement en train de faire les tests ainsi que réaliser la documentation.

Nous avons aussi bien avancé sur la rédaction de notre rapport intermédiaire.

DISCUSION	LE TROIS TIER	
Nous avons abordé pendant la réunion l'architecture trois tiers afin de bien comprendre son concept. Voir annexe de ce rapport.		
PROCHAINE RÉUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police.	À définir aux alentours de janvier

Compte rendu n°8



PST PLEIADE (EPSILON)

REUNION ORGANISEE PAR	Anthony Billette
DATE DE LA RÉUNION	24/01/2019
HEURE	De 14 h 30 à 16h
LIEU	Poste de police
TYPE DE RÉUNION	Réunion d'amélioration du projet et des objectifs
RÉDACTEURS	Théo Portier, Anthony BILLETTE, Anthony YAR
PARTICIPANTS	Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION	INTRODUCTION/PROBLEMATIQUE
	À la suite de la charge de travail que nous avons eue ces dernières semaines. Nous n'avons pas eu le temps d'avancer. L'objectif de la réunion était de présenter l'avancement du projet. Définir une version 0.2 alpha pour notre projet.
DISCUSSION	MMC
	Théo aide à la réalisation d'un script multi-os pour la MMC avec l'aide de développeur mageia venant de l'entreprise Siveo. Il est actuellement sur la partie LDAP du script qui lui pose soucis à cause du peu de connaissance qu'il a dessus.
DISCUSSION	REDEFINITION DU PROJET
	Nous avons amélioré notre premier rendu pour le projet. Comme nous pouvons le voir sur le schéma suivant.

PROCHAINE RÉUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Théo PORTIER, Simon RUFFET, Anthony YAR	Poste de police.	A définir aux alentours de fin Février

Compte rendu n°9



PST PLEIADE (EPSILON)

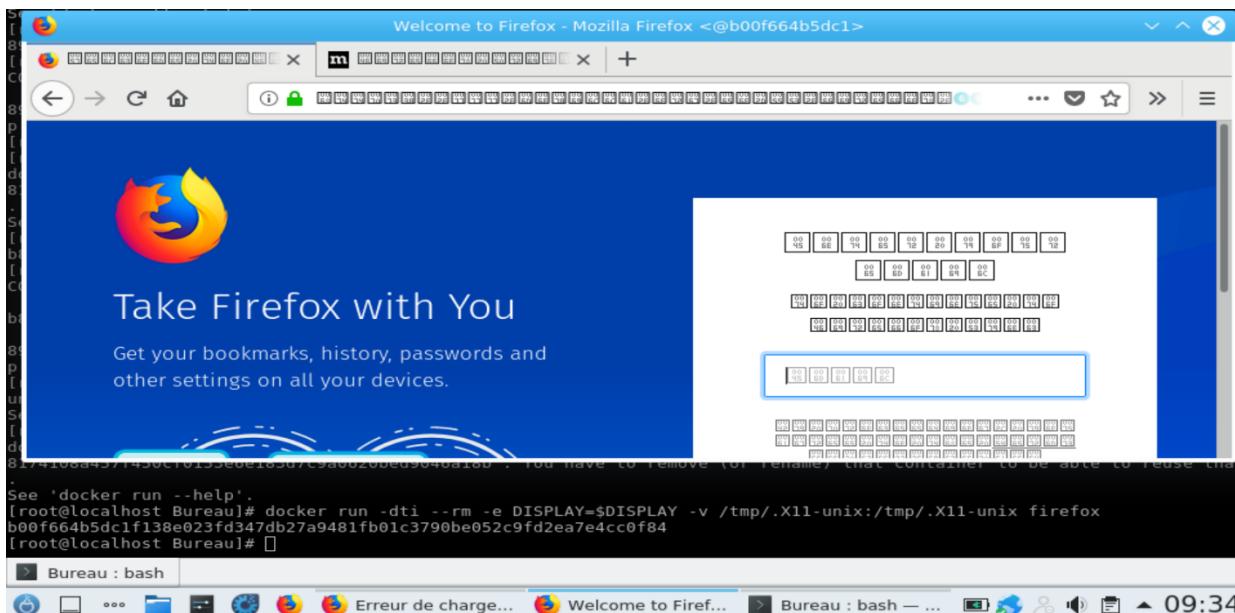
RÉUNION ORGANISÉE PAR	Anthony Billette
DATE DE LA RÉUNION	07/03/2019
HEURE	De 11 h 00 à 12h
LIEU	Poste de police
TYPE DE RÉUNION	Réunion de présentation de l'avancement du projet
RÉDACTEURS	Anthony BILLETTE
	Jean-François BELLANGER
PARTICIPANTS	Anthony BILLETTE, Jean-Baptiste PESLERBES, Simon RUFFET, Anthony YAR

Rubriques à l'ordre du jour

DISCUSSION **DIFFICULTÉS / AMÉLIORATION POUR LA SUITE**

Durant la réunion nous avons abordé l'avancement du notre PST. Nous avons démontré qu'il était possible d'utiliser le navigateur web Firefox dans un conteneur docker. Mais nous rencontrons la difficulté suivante. L'os hôte doit être de la famille de Debian pour que le conteneur Firefox se lance correctement.

Dans le cas de l'os mageia, le conteneur Firefox rencontre des difficultés d'affichage.



Nous avons présenté l'avancement sur la conteneurisation du VPN freelan.

Nous avons abordé des points de présentation pour la sallon PST / Open ESIEA.

Durant la réunion, nous avons émis l'hypothèse d'une éventuelle utilisation QubeOS pour sécurisé l'affichage déporté de notre conteneur Firefox.

Nous devons rajouter check-my-https à notre conteneur Firefox pour vérifier la connexion HTTPS.

Nous invitons Jean-François BELLANGER et son équipe à participer au salon OPEN ESIEA, le 28 mars 2019

PROCHAINE RÉUNION	LIEU	DATE ET HEURE
Membres : Jean-François BELLANGER Anthony BILLETTE, Jean-Baptiste PESLERBES, Simon RUFFET, Anthony YAR	Poste de police.	À définir aux alentours de fin Mars

Chapitre 15

Schémas du projet par ordre chronologique

On trouvera dans cette partie les schémas que nous avons réalisés pour le projet. A nos débuts, nous avons passé beaucoup de temps afin de définir l'expression du besoin final du client. Nous voulons garder une trace de nos recherches.

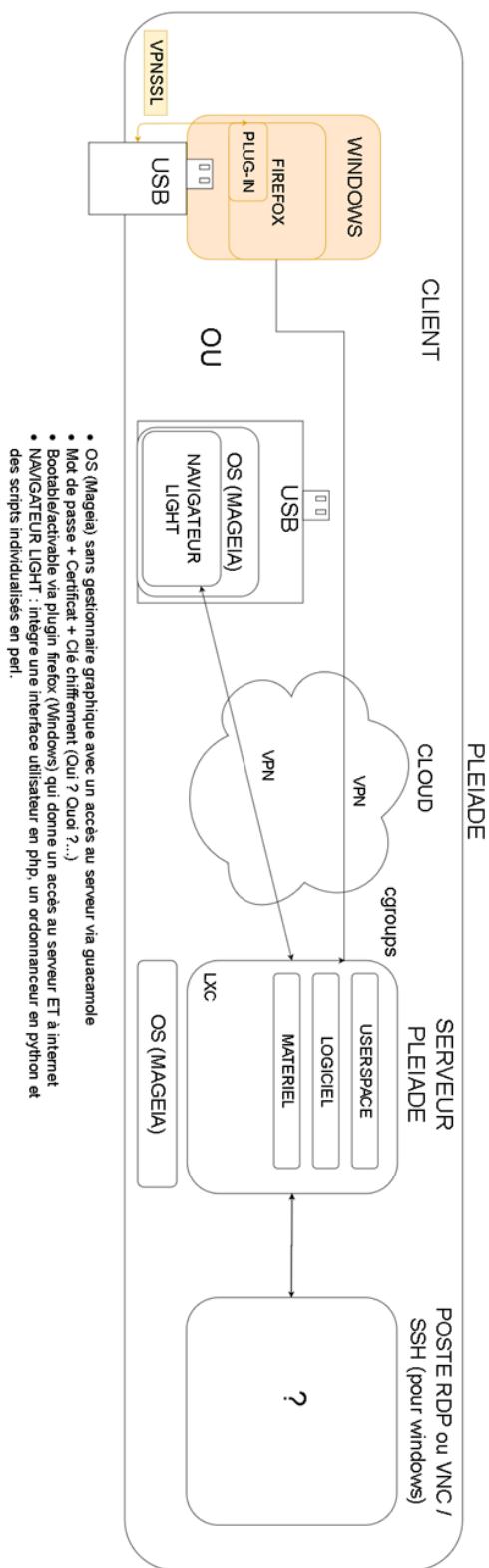


FIGURE 15.1 – Première version

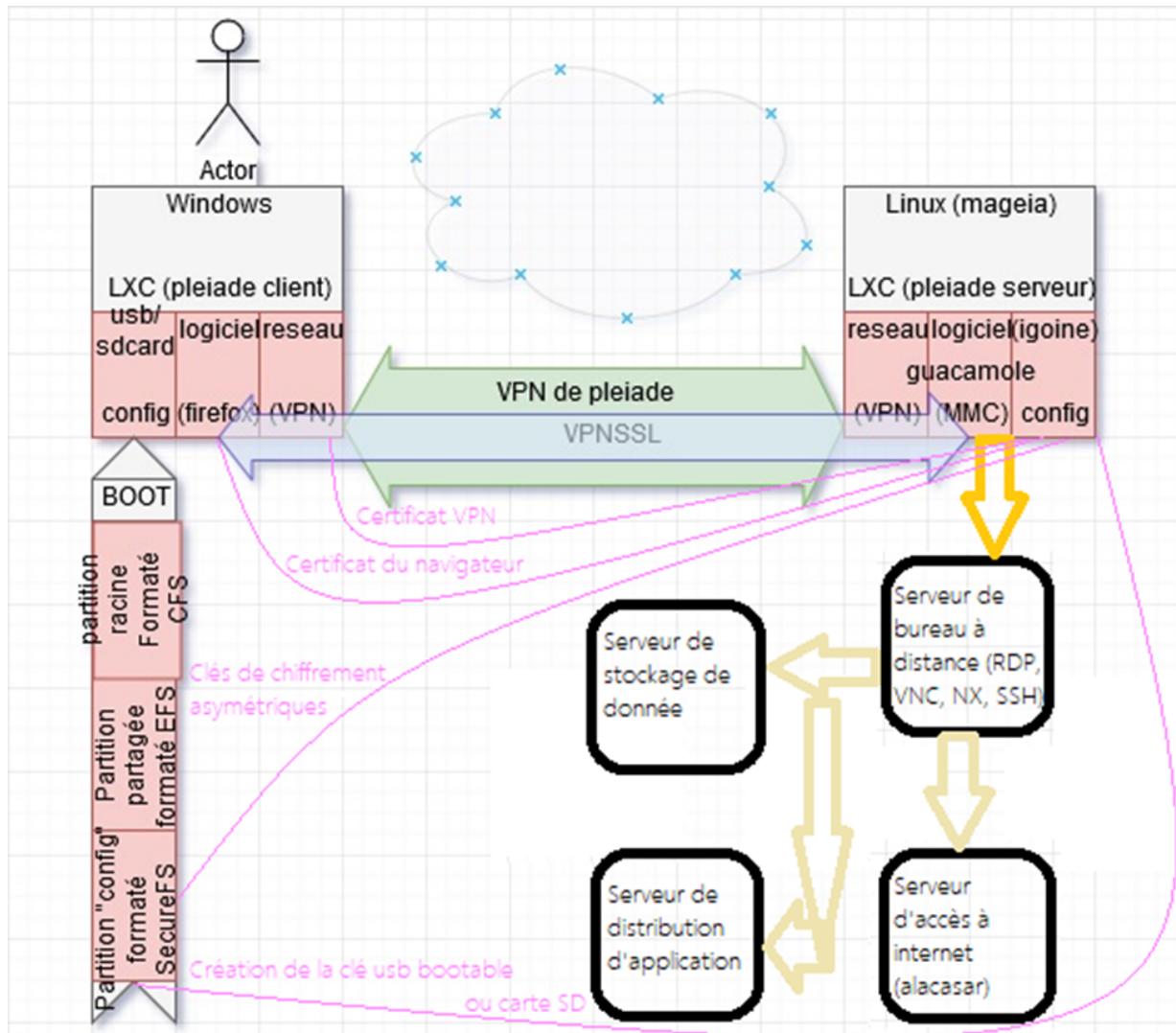


FIGURE 15.2 – Première version de Jean-François

CHAPITRE 15. SCHÉMAS DU PROJET PAR ORDRE CHRONOLOGIQUE

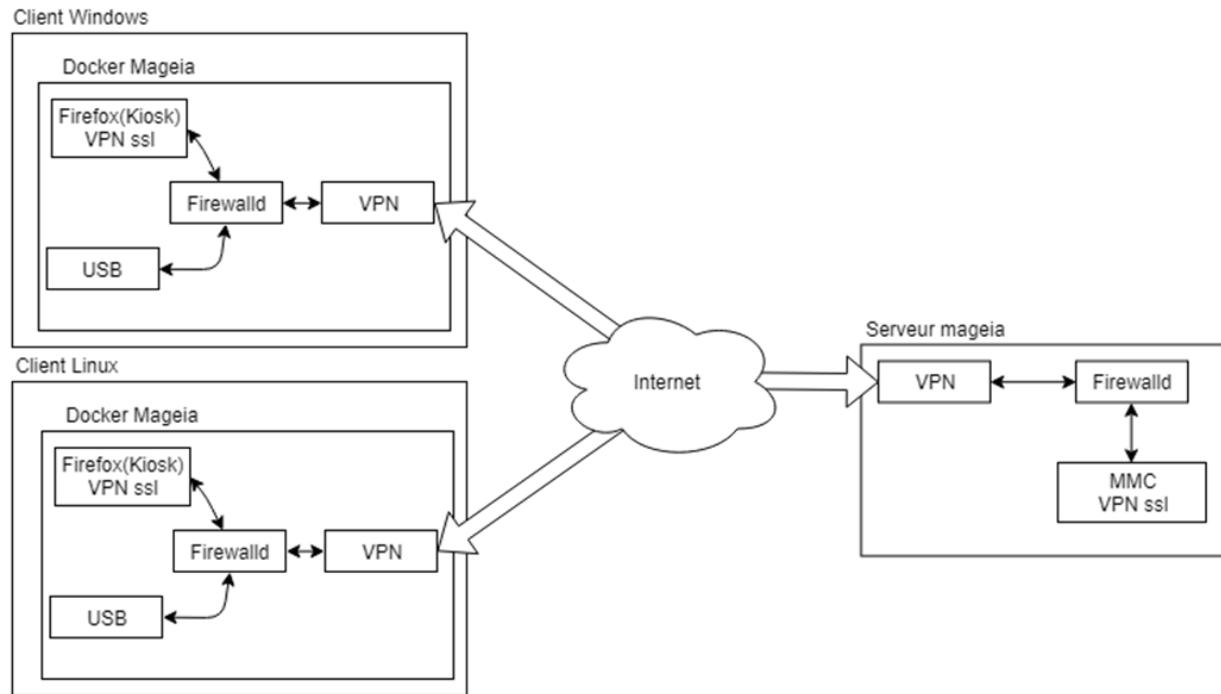


FIGURE 15.3 – Deuxième version

Côté client

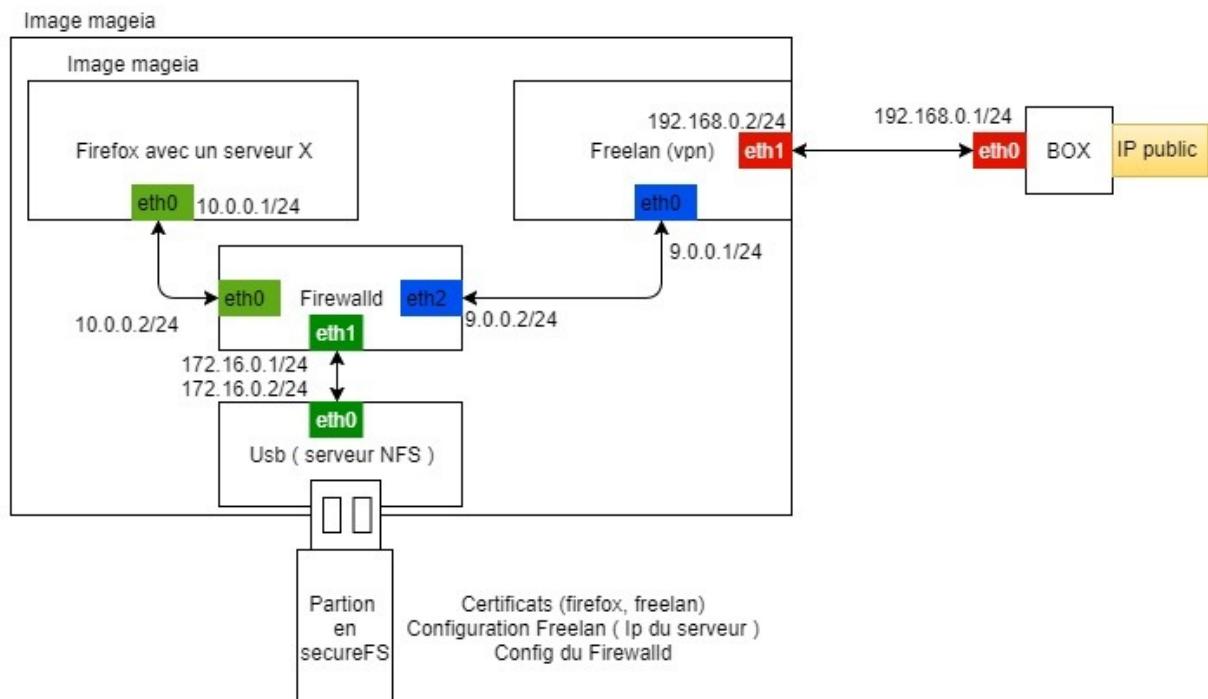


FIGURE 15.4 – Version côté client

CHAPITRE 15. SCHÉMAS DU PROJET PAR ORDRE CHRONOLOGIQUE

Affiche



PST EPSILON

Projet scientifique et technique

4A



OS NOMADE SECURISE

L'objectif du projet est de réaliser un « Environnement Portable et Sécurisé pour un Internet Libre Ouvert et Neutre » (EPSILON). Le principe est de se connecter à distance au réseau ADSL de la police national de manière sécurisée à partir d'une clé USB authentifiée.

Suiveur :

Jean-Pierre AUBIN
Jean-François BELLANGER



L'équipe :

Anthony BILLETTE
Jean-Baptiste PESLERBES
Théo PORTIER
Simon RUFFET
Anthony YAR

FIGURE 15.5 – Affiche du projet

Chapitre 16

Partie Individuelle

16.1 BILLETTE Anthony

Semestre 1 :

Dans le projet, j'ai étudié une nouvelle possibilité de conteneurisation PODMAN. Mais cette dernière s'est avérée non concluante pour nos besoins. En effet, PODMAN est un projet de conteneurisation léger, car elle n'utilise pas de démon unique démon pour lancer ces conteneurs. Elle est aussi très sécurisée, car elle se base sur SELinux (cf. partie SELinux¹ de ce rapport pour plus de détail). Après ces tests, je me suis penché sur la sécurité des conteneurs dockers et de manière générale, je me suis renseigné sur la sécurisation d'un système Linux (cf. chapitre sur la sécurisation de docker). Je suis actuellement en train de tester les possibilités qu'offre MSec. J'ai démontré que SELinux ne pouvait pas être utilisé pour notre projet et je me suis tourné sur les conseils de Jean-François BELLANGER et de Richard REY vers MSec. En Effet, Msec est utilisé pour le projet ALCASAR² dont le leader est Richard REY. Concernant la documentation, j'ai rédigé, au début du projet, la partie traitant des navigateurs web que l'on peut retrouver dans l'état de l'art. J'ai réalisé un template latex pour la rédaction de notre rapport PST. Anthony Yar a mis en forme le découpage des différents chapitres en fichier latex unique. J'ai rédigé toute la partie concernant la sécurisation de notre projet. J'ai dessiné une grosse majorité des schémas présents dans les différents documents. Organisation des réunions avec notre suiveur, mais aussi avec les membres du groupe. Mais aussi ,les différents éléments cités plus haut. La difficulté principale pour est de gérer notre équipe. Nous sommes tous des perfectionnistes dans l'âme. Je me suis rendu compte avec mon statut que j'avais moins le droit à l'erreur. Il n'est pas toujours facile de jongler entre les attentes de Jean-François BELLANGER et nos horaires de travail pour le projet. Maintenir une cohésion forte pour notre équipe

1. <https://doc.fedoraproject.org/wiki/SELinux>
2. <http://www.alcasar.net/>

même s'il y parfois des manques de motivations ponctuelles dans notre équipe, qui restent néanmoins non néfastes pour le projet.

Semestre 2 :

J'ai réalisé au second semestre la configuration de Guacamole avec l'aide d'anthony yar. J'ai pris des décisions dans le but d'obtenir un résultat pour notre projet, même si cette dernière ne remplissait pas le cahier des charge à 100%. J'ai imprimé des modèles 3D pour notre maquette final. Dans l'ensemble je trouve que ce projet, nous a apporté énormément de compétences.

16.2 PESLERBES Jean-Baptiste

Semestre 1 :

Dans ce projet j'ai été chargé de la partie conteneurisation. Dans un premier temps, il m'a été demandé de faire un état de l'art sur la technologie des conteneurs. Leur principe de fonctionnement, les différents outils qui existent, la différence entre les conteneurs et une VM... A partir de cette étude, j'ai pu choisir la technologie que l'on allait utiliser. Notre choix se portait sur LXC mais après plusieurs tests et de nouvelles attentes, nous nous sommes orienté sur Docker (qui est lxc). Concernant LXC, j'ai démontré qu'il était difficile de le faire fonctionner convenablement sur Mageia et qu'il serait long d'arriver à bout de notre projet. En effet, il y avait des problèmes pour créer des conteneurs en tant qu'utilisateur non-privilégié. De plus, il n'existe pas d'image Mageia sur LXC. En créer une sera trop long. Puis, je me suis intéressé à Docker. J'ai démontré que ce que nous voulions était faisable. J'ai rédigé toutes les parties relatives à la conteneurisation (Etat de l'art, LXC, Docker) et certains rapports de réunions. La documentation décrite ci-dessus, un dockerFile installant FreeLAN et tout le Docker en général. J'ai eu du mal à comprendre le sujet au début. J'ai perdu énormément de temps à essayer de faire fonctionner LXC sur Mageia . Concernant Docker, c'est un logiciel génial mais qui nécessite des connaissances sur toutes les facettes du système (Réseau, Variable d'environnement, processus, ...) que je n'ai pas. J'avance doucement mais sûrement et j'apprends plein de choses !

Semestre 2 :

Le semestre 2 a été moins théorique que le premier et c'était plutôt plaisant de voir ce que nous avions imaginé devant nous. Nous avons plus avancé que ce que j'avais imaginé. Dommage que nous n'ayons pas plus de temps et que le second semestre soit aussi chargé. J'ai beaucoup progresser dans beaucoup de domaines et j'ai appris plein de choses. Que du positif dans ce second semestre avec peut être un petit goût d'inachevé même si je suis plutôt content de ce qu'on rend.

16.3 PORTIER Théo

Dans le projet de cette année, j'ai réalisé plusieurs études qui m'ont permis de découvrir la partie sur laquelle je devrais travailler plus particulièrement sur le projet. Au début du projet j'ai réalisé une étude sur les OS ainsi que des recherches afin de créer un ISO. J'ai ensuite étudié la MMC et ce qu'elle impliquait (La mise en place de cette dernière, les plugins, les failles qu'elle comporte). Pour le moment j'ai réalisé plusieurs tests d'installation de la MMC afin de comprendre comment cette dernière fonctionnait. Le document que j'ai réalisé pour le moment est le test sur ce que j'ai pu faire, avec les problèmes que j'ai eus. J'ai aussi rédigé quelques rapports de réunion pour le projet. J'ai réalisé tous les tests concernant la MMC, ainsi que les recherches sur cette partie. J'ai également pris contact avec les développeurs des paquets pour la MMC afin qu'ils m'aideent à résoudre le souci que j'avais sur leurs paquets. Le fait que la base du projet était un outil qui ne fonctionnait pas au début (Pléiade) m'a démotivé pour le projet, et m'a déplu pendant un temps, cependant après la réécriture du projet, j'ai trouvé le projet plus attrayant qu'au départ. Le fait que la partie de la MMC ne fonctionne pas du tout est un point très démotivant, et c'est aussi très stressant. Cependant je pense que cela m'apportera beaucoup de savoir utiliser et manipuler une MMC pour ma future vie d'ingénieur.

16.4 RUFFET Simon

Semestre 1 :

Sur ce projet je me suis concentré sur la partie VPN. J'ai donc réalisé un état de l'art sur les différentes technologies disponible avec leurs inconvénients et avantages. Cela nous a permit de choisir le protocole le plus adapté à notre projet. Une fois la technologie trouvé, il a fallu chercher le logiciel que nous allions choisir. J'en ai donc testé plusieurs dont Wireguard et OpenVPN qui n'ont pas été concluant pour les raisons évoquées dans le rapport. J'ai réalisé toute la documentation en rapport avec Freelan. J'ai donc fait une documentation d'installation, deux méthodes pour packager Freelan, un fichier DockerFile et le fichier de configuration traduit intégralement en français. J'ai réalisé les tests des différents VPN et choisi le plus pertinent pour notre projet. J'ai réaliser une documentation d'installation pour finir par réaliser un dockerfile qui permet de créer un conteneur avec freelan d'installé à l'intérieur. Nous sommes un groupe PST qui fonctionne bien. Le sujet me plaît même si à un moment ça n'était pas le cas. En effet quand on a commencé on a été de mauvaise nouvelle en mauvaise nouvelle cependant c'est aussi un projet riche en compétences techniques et méthodologie. Je suis content que Anthony soit le meneur du groupe. Cela ma permit d'avoir une autre position dans le groupe que sur mes autres PST et ma permit de comprendre des erreurs à éviter et des manières de diriger. Pour finir nous avons un bon suiveur et un projet motivant par sa finalité.

Semestre 2 :

Le PST a été un peu plus loin que ce que je pensait au début du semestre 2. En effet, je ne pensais pas arriver à un démonstrateur sachant que nous avons eu peu de temps personnel avec le travail des autres matières. Au final la création du démonstrateur a été motivant et a permis de finir ce PST sur un bon sentiment. Même si c'est compliqué d'arriver à un démonstrateur au début du projet, cela nous aurait sûrement permis d'avancer plus vite.

16.5 YAR Anthony

Semestre 1 :

Au début de ce projet, nous pensions mettre en place un plug-in Firefox pour communiquer avec la clé USB. J'ai donc commencé par effectuer des recherches sur les plug-ins Firefox afin de rédiger l'état de l'art de ces derniers. Je me suis ensuite spécialisé dans la gestion des firewalls. J'ai démontré qu'il était possible d'utiliser les Iptables et Firewalld pour notre projet, mais que Firewalld était plus intéressant. Dans un premier temps j'ai effectué des tests avec Iptables, pour finir par choisir Firewalld. J'ai donc effectué des recherches afin de maîtriser Firewalld. J'ai rédigé toutes les parties qui concernent Firewalld. Au niveau de la documentation, j'ai également mis en place avec Anthony Billette le rapport sous Latex, je me suis notamment occupé de réorganiser la partie annexe. La principale difficulté que j'ai rencontrée était le manque de documentation sur internet pour utiliser les technologies sur MAGEIA. J'ai notamment perdu beaucoup de temps à trouver ce qui bloquait le lancement au démarrage des Iptables et de Firewalld.

Semestre 2 :

Au deuxième semestre, j'ai participé à la gestion du serveur guacamole avec Anthony Billette. Ce deuxième semestre a été cours ce qui nous a posé quelques problèmes de gestion de temps. Il a cependant été très intéressant et beaucoup plus motivant puisque nous avons abouti à quelque chose de graphique.

Table des figures

2.1	Salle	14
2.2	Tableau post-its	15
2.3	Gantt	15
3.1	Principe du VPN	17
3.2	Encapsulation	17
3.3	Structure d'un plug-in	20
3.4	Containers vs VMs	21
3.5	Screenshot de la documentation Ubuntu 1/2	27
3.6	Screenshot de la documentation Ubuntu 2/2	28
3.7	Architecture 3 tiers pour une application web	31
3.8	Schéma de la défense en profondeur pour le projet EPSILON	31
4.1	Schéma de la MMC	38
5.1	-list-all	42
5.2	MSEC	43
5.3	Lynis résultat	44
5.4	Image_guacamole	48
5.5	Image_guacamole2	48
6.1	Avancement schématisé	50
7.1	La première étape	54
7.2	La deuxième étape	54

7.3	La troisième étape	55
7.4	La quatrième étape	55
9.1	script_firewalld	92
9.2	rc.local	92
10.1	Arborescence des forks linux coté Mageia	95
12.1	lxc-sub	100
12.2	test_root	101
12.3	test_lxc1	102
12.4	test_lxc2	102
12.5	test_user_lxc	102
12.6	lxc-ls	103
12.7	urpmi	103
12.8	Cgroups	104
12.9	lxc_script	104
12.10	config_lxc	105
12.11	Activation des dépôts	107
12.12	Installation	107
12.13	Lancement de notre image	107
12.14	Lancement d'un conteneur sans SELinux	108
12.15	Erreur lors du lancement de l'agent de la MMC	109
12.16	Journal du service de l'agent de la MMC	109
12.17	Nouvelle erreur dans le journal de l'agent	109
12.18	Lancement du script de l'annuaire LDAP	110
12.19	Erreur lors de la connexion à la MMC	111
12.20	Erreur lors du lancement du conteneur contenant une MMC	112
12.21	list-all_drop	113
12.22	ping1	113
12.23	ping2	114
12.24	ping3	114

12.25list_all_ssh	115
12.26ping4	115
12.27ssh1	115
12.28ping+ssh	116
12.29schéma port-forwarding	117
12.30image ifcfg	117
13.1 schemaPresentation	119
13.2 Maquette	120
15.1 Première version	144
15.2 Première version de Jean-François	145
15.3 Deuxième version	146
15.4 Version côté client	146
15.5 Affiche du projet	147
15.6 BILLETTE Anthony	148
15.7 PESLERBES Jean-Baptiste	148
15.8 PORTIER Théo	148
15.9 RUFFET Simon	148
15.10YAR Anthony	148

Troisième partie

Complément d'annexes

Chapitre 17

Liste des booléens sur SELinux

- **acct_disable_trans** (SELinux Service Protection)
Disable SELinux protection for acct daemon
- **allow_cvs_read_shadow** (CVS)
Allow cvs daemon to read shadow
- **allow_daemons_dump_core** (Admin)
Allow all daemons to write corefiles to /.
- **allow_daemons_use_tty** (Admin)
Allow all daemons the ability to use unallocated ttys.
- **allow_execheap** (Memory Protection)
Allow unconfined executables to make their heap memory executable. Doing this is a really bad idea. Probably indicates a badly coded executable, but could indicate an attack. This executable should be reported in bugzilla
- **allow_execmem** (Memory Protection)
Allow unconfined executables to map a memory region as both executable and writable, this is dangerous and the executable should be reported in bugzilla
- **allow_execmod** (Memory Protection)
Allow all unconfined executables to use libraries requiring text relocation that are not labeled textrel_shlib_t
- **allow_execstack** (Memory Protection)
Allow unconfined executables to make their stack executable. This should never, ever be necessary. Probably indicates a badly coded executable, but could indicate an attack. This executable should be reported in bugzilla
- **allow_ftpd_full_access** (FTP)
Allow ftpd to full access to the system
- **allow_ftpd_anon_write** (FTP)
Allow ftpd to upload files to directories labeled public_content_rw_t

- **allow_ftpd_use_cifs** (FTP)
Allow ftp servers to use cifs used for public file transfer services.
- **allow_ftpd_use_nfs** (FTP)
Allow ftp servers to use nfs used for public file transfer services.
- **allow_gpg_execstack** (Memory Protection)
Allow gpg executable stack
- **allow_gssd_read_tmp** (NFS)
Allow gssd to read temp directory.
- **allow_httpd_anon_write** (HTTPD Service)
Allow httpd daemon to write files in directories labeled public_content_rw_t
- **allow_httpd_mod_auth_pam** (HTTPD Service)
Allow Apache to use mod_auth_pam.
- **allow_httpd_sys_script_anon_write** (HTTPD Service)
Allow httpd scripts to write files in directories labeled public_content_rw_t
- **allow_java_execstack** (Memory Protection)
Allow java executable stack
- **allow_kerberos** (Kerberos)
Allow daemons to use kerberos files
- **allow_mount_anyfile** (Mount)
Allow mount to mount any file
- **allow_mounton_anydir** (Mount)
Allow mount to mount any dir
- **allow_mplayer_execstack** (Memory Protection)
Allow mplayer executable stack
- **allow_nfsd_anon_write** (NFS)
Allow nfs servers to modify public files used for public file transfer services.
- **allow_polyinstantiation** (Polyinstantiation)
Enable polyinstantiated directory support.
- **allow_ptrace** (Compatibility)
Allow sysadm_t to debug or ptrace applications
- **allow_rsync_anon_write** (rsync)
Allow rsync to write files in directories labeled public_content_rw_t
- **allow_smbd_anon_write** (Samba)
Allow Samba to write files in directories labeled public_content_rw_t
- **allow_ssh_keysign** (SSH)
Allow ssh to run ssh-signkey

- **allow_unconfined_execmem_dyntrans** (Memory Protection)
Allow unconfined to dyntrans to unconfined_execmem
- **allow_user_mysql_connect** (Databases)
Allow user to connect to mysql socket
- **allow_user_postgresql_connect** (Databases)
Allow user to connect to postgres socket
- **allow_write_xshm** (XServer)
Allow clients to write to X shared memory
- **allow_ypbind** (NIS)
Allow daemons to run with NIS
- **allow_zebra_write_config** (Zebra)
Allow zebra daemon to write its configuration files
- **amanda_disable_trans** (SELinux Service Protection)
Disable SELinux protection for amanda
- **amavis_disable_trans** (SELinux Service Protection)
Disable SELinux protection for amavis
- **apmd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for apmd daemon
- **arpwatch_disable_trans** (SELinux Service Protection)
Disable SELinux protection for arpwatch daemon
- **auditd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for auditd daemon
- **automount_disable_trans** (Mount)
Disable SELinux protection for automount daemon
- **avahi_disable_trans** (SELinux Service Protection)
Disable SELinux protection for avahi
- **bluetooth_disable_trans** (SELinux Service Protection)
Disable SELinux protection for bluetooth daemon
- **canna_disable_trans** (SELinux Service Protection)
Disable SELinux protection for canna daemon
- **cardmgr_disable_trans** (SELinux Service Protection)
Disable SELinux protection for cardmgr daemon
- **ccs_disable_trans** (SELinux Service Protection)
Disable SELinux protection for Cluster Server
- **cdrecord_read_content** (User Prvs)
Allow cdrecord to read various content. nfs, samba, removable devices, user temp and untrusted content files

- **ciped_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ciped daemon
- **clamd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for clamd daemon
- **clamscan_disable_trans** (SELinux Service Protection)
Disable SELinux protection for clamscan
- **clvmd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for clvmd
- **comsat_disable_trans** (SELinux Service Protection)
Disable SELinux protection for comsat daemon
- **courier_authdaemon_disable_trans** (SELinux Service Protection)
Disable SELinux protection for courier daemon
- **courier_pcp_disable_trans** (SELinux Service Protection)
Disable SELinux protection for courier daemon
- **courier_pop_disable_trans** (SELinux Service Protection)
Disable SELinux protection for courier daemon
- **courier_sqwebmail_disable_trans** (SELinux Service Protection)
Disable SELinux protection for courier daemon
- **courier_tcpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for courier daemon
- **cpucontrol_disable_trans** (SELinux Service Protection)
Disable SELinux protection for cpucontrol daemon
- **cpuspeed_disable_trans** (SELinux Service Protection)
Disable SELinux protection for cpuspeed daemon
- **cron_can_relabel** (Cron)
Allow system cron jobs to relabel filesystem for restoring file contexts.
- **crond_disable_trans** (Cron)
Disable SELinux protection for crond daemon
- **cupsd_config_disable_trans** (Printing)
Disable SELinux protection for cupsd backend server
- **cupsd_disable_trans** (Printing)
Disable SELinux protection for cupsd daemon
- **cupsd_lpd_disable_trans** (Printing)
Disable SELinux protection for cupsd_lpd
- **cvs_disable_trans** (CVS)
Disable SELinux protection for cvs daemon

- **cyrus_disable_trans** (SELinux Service Protection)
Disable SELinux protection for cyrus daemon
- **dbskkd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dbskkd daemon
- **dbusd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dbusd daemon
- **dccd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dccd
- **dccifd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dccifd
- **dccm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dccm
- **ddt_client_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ddt daemon
- **devfsd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for devfsd daemon
- **dhcpc_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dhcpc daemon
- **dhcpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dhcpd daemon
- **dictd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dictd daemon
- **direct_sysadm_daemon** (Admin)
Allow sysadm_t to directly start daemons
- **disable_evolution_trans** (Web Applications)
Disable SELinux protection for Evolution
- **disable_games_trans** (Games)
Disable SELinux protection for games
- **disable_mozilla_trans** (Web Applications)
Disable SELinux protection for the web browsers
- **disable_thunderbird_trans** (Web Applications)
Disable SELinux protection for Thunderbird
- **distccd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for distccd daemon
- **dmesg_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dmesg daemon

- **dnsmasq_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dnsmasq daemon
- **dovecot_disable_trans** (SELinux Service Protection)
Disable SELinux protection for dovecot daemon
- **entropyd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for entropyd daemon
- **fcron_crond** (Cron)
Enable extra rules in the cron domain to support fcron.
- **fetchmail_disable_trans** (SELinux Service Protection)
Disable SELinux protection for fetchmail
- **fingerd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for fingerd daemon
- **freshclam_disable_trans** (SELinux Service Protection)
Disable SELinux protection for freshclam daemon
- **fsdaemon_disable_trans** (SELinux Service Protection)
Disable SELinux protection for fsdaemon daemon
- **ftpd_disable_trans** (FTP)
Disable SELinux protection for ftpd daemon
- **ftpd_is_daemon** (FTP)
Allow ftpd to run directly without inetd
- **ftp_home_dir** (FTP)
Allow ftp to read/write files in the user home directories
- **global_ssp** (Admin)
This should be enabled when all programs are compiled with ProPolice/SSP stack smashing protection. All domains will be allowed to read from /dev/urandom.
- **gpm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for gpm daemon
- **gssd_disable_trans** (NFS)
Disable SELinux protection for gss daemon
- **hald_disable_trans** (SELinux Service Protection)
Disable SELinux protection for hal daemon
- **hide_broken_symptoms** (Compatibility)
Do not audit things that we know to be broken but which are not security risks
- **hostname_disable_trans** (SELinux Service Protection)
Disable SELinux protection for hostname daemon
- **hotplug_disable_trans** (SELinux Service Protection)
Disable SELinux protection for hotplug daemon

- **howl_disable_trans** (SELinux Service Protection)
Disable SELinux protection for howl daemon
- **hplip_disable_trans** (Printing)
Disable SELinux protection for cups hplip daemon
- **httpd_builtin_scripting** (HTTPD Service)
Allow HTTPD to support built-in scripting
- **httpd_can_network_connect_db** (HTTPD Service)
Allow HTTPD scripts and modules to network connect to databases.
- **httpd_can_network_connect** (HTTPD Service)
Allow HTTPD scripts and modules to connect to the network.
- **httpd_can_network_relay** (HTTPD Service)
Allow httpd to act as a relay.
- **httpd_disable_trans** (HTTPD Service)
Disable SELinux protection for httpd daemon
- **httpd_enable_cgi** (HTTPD Service)
Allow HTTPD cgi support
- **httpd_enable_ftp_server** (HTTPD Service)
Allow HTTPD to run as a ftp server
- **httpd_enable_homedirs** (HTTPD Service)
Allow HTTPD to read home directories
- **httpd_rotatelogs_disable_trans** (SELinux Service Protection)
Disable SELinux protection for httpd rotatelog
- **httpd_ssi_exec** (HTTPD Service)
Allow HTTPD to run SSI executables in the same domain as system CGI scripts.
- **httpd_suexec_disable_trans** (HTTPD Service)
Disable SELinux protection for http suexec
- **httpd_tty_comm** (HTTPD Service)
Unify HTTPD to communicate with the terminal. Needed for handling certificates.
- **httpd_unified** (HTTPD Service)
Unify HTTPD handling of all content files.
- **hwclock_disable_trans** (SELinux Service Protection)
Disable SELinux protection for hwclock daemon
- **i18n_input_disable_trans** (SELinux Service Protection)
Disable SELinux protection for i18n daemon
- **imazesrv_disable_trans** (SELinux Service Protection)
Disable SELinux protection for imazesrv daemon

- **inetd_child_disable_trans** (SELinux Service Protection)
Disable SELinux protection for inetd child daemons
- **inetd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for inetd daemon
- **innd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for innd daemon
- **iptables_disable_trans** (SELinux Service Protection)
Disable SELinux protection for iptables daemon
- **ircd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ircd daemon
- **irqbalance_disable_trans** (SELinux Service Protection)
Disable SELinux protection for irqbalance daemon
- **iscsid_disable_trans** (SELinux Service Protection)
Disable SELinux protection for iscsi daemon
- **jabberd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for jabberd daemon
- **kadmind_disable_trans** (Kerberos)
Disable SELinux protection for kadmind daemon
- **klogd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for klogd daemon
- **krb5kdc_disable_trans** (Kerberos)
Disable SELinux protection for krb5kdc daemon
- **ktalkd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ktalk daemons
- **kudzu_disable_trans** (SELinux Service Protection)
Disable SELinux protection for kudzu daemon
- **locate_disable_trans** (SELinux Service Protection)
Disable SELinux protection for locate daemon
- **lpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for lpd daemon
- **lrrd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for lrrd daemon
- **lvm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for lvm daemon
- **mailman_mail_disable_trans** (SELinux Service Protection)
Disable SELinux protection for mailman

- **mail_read_content** (Web Applications)
Allow evolution and thunderbird to read user files
- **mdadm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for mdadm daemon
- **monopd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for monopd daemon
- **mozilla_read_content** (Web Applications)
Allow the mozilla browser to read user files
- **mrtg_disable_trans** (SELinux Service Protection)
Disable SELinux protection for mrtg daemon
- **mysqld_disable_trans** (Databases)
Disable SELinux protection for mysqld daemon
- **nagios_disable_trans** (SELinux Service Protection)
Disable SELinux protection for nagios daemon
- **named_disable_trans** (Name Service)
Disable SELinux protection for named daemon
- **named_write_master_zones** (Name Service)
Allow named to overwrite master zone files
- **nessusd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for nessusd daemon
- **NetworkManager_disable_trans** (SELinux Service Protection)
Disable SELinux protection for NetworkManager
- **nfsd_disable_trans** (NFS)
Disable SELinux protection for nfsd daemon
- **nfs_export_all_ro** (NFS)
Allow NFS to share any file/directory read only
- **nfs_export_all_rw** (NFS)
Allow NFS to share any file/directory read/write
- **nmbd_disable_trans** (Samba)
Disable SELinux protection for nmbd daemon
- **nrpe_disable_trans** (SELinux Service Protection)
Disable SELinux protection for nrpe daemon
- **nscd_disable_trans** (Name Service)
Disable SELinux protection for nscd daemon
- **nsd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for nsd daemon

- **ntpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ntpd daemon
- **oddjob_disable_trans** (SELinux Service Protection)
Disable SELinux protection for oddjob
- **oddjob_mkhomedir_disable_trans** (SELinux Service Protection)
Disable SELinux protection for oddjob_mkhomedir
- **openvpn_disable_trans** (SELinux Service Protection)
Disable SELinux protection for openvpn daemon
- **pam_console_disable_trans** (SELinux Service Protection)
Disable SELinux protection for pam daemon
- **pegasus_disable_trans** (SELinux Service Protection)
Disable SELinux protection for pegasus
- **perdition_disable_trans** (SELinux Service Protection)
Disable SELinux protection for perdition daemon
- **portmap_disable_trans** (SELinux Service Protection)
Disable SELinux protection for portmap daemon
- **portslave_disable_trans** (SELinux Service Protection)
Disable SELinux protection for portsclave daemon
- **postfix_disable_trans** (SELinux Service Protection)
Disable SELinux protection for postfix
- **postgresql_disable_trans** (Databases)
Disable SELinux protection for postgresql daemon
- **pppd_can_insmod** (pppd)
Allow pppd daemon to insert modules into the kernel
- **pppd_disable_trans** (pppd)
Disable SELinux protection for pppd daemon
- **pppd_disable_trans** (pppd)
Disable SELinux protection for the mozilla ppp daemon
- **pppd_for_user** (pppd)
Allow pppd to be run for a regular user.
- **pptp_disable_trans** (SELinux Service Protection)
Disable SELinux protection for pptp
- **prelink_disable_trans** (SELinux Service Protection)
Disable SELinux protection for prelink daemon
- **privoxy_disable_trans** (SELinux Service Protection)
Disable SELinux protection for privoxy daemon

- **ptal_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ptal daemon
- **pxe_disable_trans** (SELinux Service Protection)
Disable SELinux protection for pxe daemon
- **pyzord_disable_trans** (SELinux Service Protection)
Disable SELinux protection for pyzord
- **quota_disable_trans** (SELinux Service Protection)
Disable SELinux protection for quota daemon
- **radiusd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for radiusd daemon
- **radvd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for radvd daemon
- **rdisc_disable_trans** (SELinux Service Protection)
Disable SELinux protection for rdisc
- **readahead_disable_trans** (SELinux Service Protection)
Disable SELinux protection for readahead
- **read_default_t** (Admin)
Allow programs to read files in non-standard locations default_t
- **read_untrusted_content** (Web Applications)
Allow programs to read untrusted content without relabel
- **restorecond_disable_trans** (SELinux Service Protection)
Disable SELinux protection for restorecond
- **rhgb_disable_trans** (SELinux Service Protection)
Disable SELinux protection for rhgb daemon
- **ricci_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ricci
- **ricci_modclusterd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ricci_modclusterd
- **rlogind_disable_trans** (SELinux Service Protection)
Disable SELinux protection for rlogind daemon
- **rpcd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for rpcd daemon
- **rshd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for rshd
- **rsync_disable_trans** (rsync)
Disable SELinux protection for rsync daemon

- **run_ssh_inetd** (SSH)
Allow ssh to run from inetd instead of as a daemon
- **samba_enable_home_dirs** (Samba)
Allow Samba to share users home directories
- **samba_share_nfs** (Samba)
Allow Samba to share nfs directories
- **allow_saslauthd_read_shadow** (SASL authentication server)
Allow sasl authentication server to read /etc/shadow
- **saslauthd_disable_trans** (SASL authentication server)
Disable SELinux protection for saslauthd daemon
- **scannerdaemon_disable_trans** (SELinux Service Protection)
Disable SELinux protection for scannerdaemon daemon
- **secure_mode** (Admin)
Do not allow transition to sysadm_t, sudo and su effected
- **secure_mode_insmod** (Admin)
Do not allow any processes to load kernel modules
- **secure_mode_policyload** (Admin)
Do not allow any processes to modify kernel SELinux policy
- **sendmail_disable_trans** (SELinux Service Protection)
Disable SELinux protection for sendmail daemon
- **setrans_disable_trans** (SELinux Service Protection)
Disable SELinux protection for setrans
- **setroubleshoot_disable_trans** (SELinux Service Protection)
Disable SELinux protection for setroublesoot daemon
- **slapd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for slapd daemon
- **slrnpull_disable_trans** (SELinux Service Protection)
Disable SELinux protection for slrnpull daemon
- **smbd_disable_trans** (Samba)
Disable SELinux protection for smbd daemon
- **snmpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for snmpd daemon
- **snort_disable_trans** (SELinux Service Protection)
Disable SELinux protection for snort daemon
- **soundd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for soundd daemon

- **sound_disable_trans** (SELinux Service Protection)
Disable SELinux protection for sound daemon
- **spamassassin_can_network** (Spam Assassin)
Allow Spam Assasin daemon network access
- **spamd_disable_trans** (spam Protection)
Disable SELinux protection for spamd daemon
- **spamd_enable_home_dirs** (spam Protection)
Allow spamd to access home directories
- **spammassassin_can_network** (spam Protection)
Allow spammassassin to access the network
- **speedmgmt_disable_trans** (SELinux Service Protection)
Disable SELinux protection for speedmgmt daemon
- **squid_connect_any** (Squid)
Allow squid daemon to connect to the network
- **squid_disable_trans** (Squid)
Disable SELinux protection for squid daemon
- **ssh_keygen_disable_trans** (SSH)
Disable SELinux protection for ssh daemon
- **ssh_sysadm_login** (SSH)
Allow ssh logins as sysadm_r :sysadm_t
- **staff_read_sysadm_file** (Admin)
Allow staff_r users to search the sysadm home dir and read files such as /.bashrc
- **stunnel_disable_trans** (Universal SSL tunnel)
Disable SELinux protection for stunnel daemon
- **stunnel_is_daemon** (Universal SSL tunnel)
Allow stunnel daemon to run as standalone, outside of xinetd
- **swat_disable_trans** (SELinux Service Protection)
Disable SELinux protection for swat daemon
- **sxid_disable_trans** (SELinux Service Protection)
Disable SELinux protection for sxid daemon
- **syslogd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for syslogd daemon
- **system_crond_disable_trans** (SELinux Service Protection)
Disable SELinux protection for system cron jobs
- **tcpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for tcp daemon

- **telnetd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for telnet daemon
- **tftpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for tftpd daemon
- **transproxy_disable_trans** (SELinux Service Protection)
Disable SELinux protection for transproxy daemon
- **udev_disable_trans** (SELinux Service Protection)
Disable SELinux protection for udev daemon
- **uml_switch_disable_trans** (SELinux Service Protection)
Disable SELinux protection for uml daemon
- **unlimitedInetd** (Admin)
Allow xinetd to run unconfined, including any services it starts that do not have a domain transition explicitly defined.
- **unlimitedRC** (Admin)
Allow rc scripts to run unconfined, including any daemon started by an rc script that does not have a domain transition explicitly defined.
- **unlimitedRPM** (Admin)
Allow rpm to run unconfined.
- **unlimitedUtils** (Admin)
Allow privileged utilities like hotplug and insmod to run unconfined.
- **updfstab_disable_trans** (SELinux Service Protection)
Disable SELinux protection for updfstab daemon
- **uptimed_disable_trans** (SELinux Service Protection)
Disable SELinux protection for uptimed daemon
- **use_lpd_server** (Printing)
Use lpd server instead of cups
- **use_nfs_home_dirs** (NFS)
Support NFS home directories
- **user_canbe_sysadm** (User Privs)
Allow user_r to reach sysadm_r via su, sudo, or userhelper. Otherwise, only staff_r can do so.
- **user_can_mount** (Mount)
Allow users to execute the mount command
- **user_direct_mouse** (User Privs)
Allow regular users direct mouse access only allow the X server
- **user_dmesg** (User Privs)
Allow users to run the dmesg command

- **user_net_control** (User Privilages)
Allow users to control network interfaces also needs USERCTL=true
- **user_ping** (User Privilages)
Allow normal user to execute ping
- **user_rw_noextattrfile** (User Privilages)
Allow user to r/w noextattrfile FAT, CDROM, FLOPPY
- **user_rw_usb** (User Privilages)
Allow users to rw usb devices
- **user_tcp_server** (User Privilages)
Allow users to run TCP servers bind to ports and accept connection from the same domain and outside users disabling this forces FTP passive mode and may change other protocols
- **user_ttyfile_stat** (User Privilages)
Allow user to stat ttyfiles
- **use_samba_home_dirs** (Samba)
Allow users to login with CIFS home directories
- **uucpd_disable_trans** (SELinux Service Protection)
Disable SELinux protection for uucpd daemon
- **vmware_disable_trans** (SELinux Service Protection)
Disable SELinux protection for vmware daemon
- **watchdog_disable_trans** (SELinux Service Protection)
Disable SELinux protection for watchdog daemon
- **winbind_disable_trans** (Samba)
Disable SELinux protection for winbind daemon
- **write_untrusted_content** (Web Applications)
Allow web applications to write untrusted content to disk implies read
- **xdm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for xdm daemon
- **xdm_sysadm_login** (XServer)
Allow xdm logins as sysadm_r :sysadm_t
- **xend_disable_trans** (SELinux Service Protection)
Disable SELinux protection for xen daemon
- **xen_use_raw_disk** (XEN)
Allow xen to read/write physical disk devices
- **xfs_disable_trans** (SELinux Service Protection)
Disable SELinux protection for xfs daemon

- **xm_disable_trans** (SELinux Service Protection)
Disable SELinux protection for xen constrol
- **ypbind_disable_trans** (NIS)
Disable SELinux protection for ypbind daemon
- **ypasswdd_disable_trans** (NIS)
Disable SELinux protection for NIS Password Daemon
- **ypserv_disable_trans** (SELinux Service Protection)
Disable SELinux protection for ypserv daemon
- **ypxfr_disable_trans** (NIS)
Disable SELinux protection for NIS Transfer Daemon
- **zebra_disable_trans** (SELinux Service Protection)
Disable SELinux protection for zebra daemon
- **httpd_use_cifs** (HTTPD Service)
Allow httpd to access samba/cifs file systems.
- **httpd_use_nfs** (HTTPD Service)
Allow httpd to access nfs file systems.
- **samba_domain_controller** (Samba)
Allow samba to act as the domain controller, add users, groups and change passwords
- **samba_export_all_ro** (Samba)
Allow Samba to share any file/directory read only
- **samba_export_all_rw** (Samba)
Allow Samba to share any file/directory read/write
- **webadm_manage_users_files** (HTTPD Service)
Allow httpd to access nfs file systems.
- **webadm_read_users_files** (HTTPD Service)
Allow httpd to access nfs file systems.

Chapitre 18

MSEC règles et paramètres

ENABLE_IP_SPOOFING_PROTECTION Activer/Désactiver la protection contre l’usurpation de la résolution des noms.

MAIL_EMPTY_CONTENT Permet l’envoi de rapports de courrier vide.

ACCEPT_BROADCASTED_ICMP_ECHO Accepter/Refuser l’écho ICMP diffusé.

ALLOW_XSERVER_TO_LISTEN L’argument spécifie si les clients sont autorisés à se connecter au serveur X sur le port TCP 6000 ou non.

CHECK_CHKROOTKIT Permet de vérifier les rootkits connus en utilisant chkrootkit.

CHECK_SUID_ROOT Permet de vérifier les ajouts/suppressions de fichiers racine suid.

ENABLE_AT_CRONTAB Activer/Désactiver crontab et at pour les utilisateurs. Placez les utilisateurs autorisés dans /etc/cron.allow et /etc/at.allow (voir man at(1) et crontab(1)).

ACCEPT_BOGUS_ERROR_RESPONSES Accepter/refuser de faux messages d’erreur IPv4.

CHECK_SUID_MD5 Permet la vérification de la somme de contrôle pour les fichiers suid.

MAIL_USER Définit l’email pour recevoir des notifications de sécurité.

ALLOW_AUTOLOGIN Autoriser/favoriser l’autologin.

ENABLE_PAM_WHEEL_FOR_SU Activer su uniquement à partir des membres du groupe de roues ou autoriser su à partir de n’importe quel utilisateur...

CREATE_SERVER_LINK Crée le lien symbolique */etc/security/msec/server* pour pointer vers */etc/security/msec/server.SERVER_LEVEL*. Le */etc/security/msec/server* est utilisé par chkconfig –add pour décider d’ajouter un service s’il est présent dans le fichier pendant l’installation des paquets.

SHELL_TIMEOUT Définit le délai d’attente de l’interpréteur de commandes. Une va-

leur de zéro signifie qu'il n'y a pas de délai d'attente.

CHECK_USER_FILES Permet de vérifier les permissions sur les fichiers des utilisateurs qui ne devraient pas être la propriété de quelqu'un d'autre, ou qui ne devraient pas être inscriptibles.

CHECK_SHADOW Permet de vérifier les mots de passe vides.

ENABLE_PASSWORD Utilisez un mot de passe pour authentifier les utilisateurs. Soyez EXTRÊMEMENT prudent lorsque vous désactivez les mots de passe, car cela rendra la machine COMPLÈTEMENT vulnérable.

WIN_PARTS_UMASK Définit l'option umask pour le montage des partitions VFAT et NTFS. Une valeur de None signifie umask par défaut.

CHECK_OPEN_PORT Permet de vérifier les ports réseau ouverts.

ENABLE_LOG_STRANGE_PACKETS Activer/Désactiver la journalisation des paquets IPv4 étranges.

CHECK_RPM Permet la vérification des paquets installés.

MAIL_WARN Permet la soumission des résultats de sécurité par courriel.

PASSWORD_LENGTH Définit la longueur minimale du mot de passe et le nombre minimal de chiffres et le nombre minimal de lettres majuscules.

ROOT_UMASK Définit l'umask racine.

CHECK_SGID Permet de vérifier les ajouts/suppressions de fichiers sgid.

CHECK_PROMISC Activer/Désactiver le contrôle de promiscuité des cartes ethernet.

ALLOW_X_CONNECTIONS Autoriser/Forbattre les connexions X. Arguments acceptés : oui (toutes les connexions sont autorisées), local (connexion locale uniquement), non (aucune connexion).

CHECK_WRITABLE Permet de vérifier les fichiers/répertoires qui peuvent être écrits par tout le monde.

ALLOW_X_CONNECTIONS Autoriser/Forbattre les connexions X. Arguments acceptés : oui (toutes les connexions sont autorisées), local (connexion locale uniquement), non (aucune connexion).

ENABLE_CONSOLE_LOG Activer/Désactiver les rapports syslog sur le terminal de la console 12.

ENABLE_DNS_SPOOFING_PROTECTION Activer/Désactiver la protection contre l'usurpation d'adresse IP.

BASE_LEVEL Définit le niveau de sécurité de base sur lequel se base la configuration actuelle.

CHECK_PERMS Permet la vérification périodique des permissions pour les fichiers système.

SHELL_HISTORY_SIZE Définit la taille de l'historique des commandes de l'interpréteur de commandes. Une valeur de -1 signifie illimité.

ALLOW_REBOOT Autoriser/supprimer le redémarrage et l'arrêt du système pour les utilisateurs locaux.

SYSLOG_WARN Permet l'enregistrement dans le journal système.

CHECK_SHOSTS Permet de vérifier les options dangereuses dans les fichiers *.rhosts/.shosts* des utilisateurs.

CHECK_PASSWD Permet les contrôles liés aux mots de passe, tels que les mots de passe vides et les comptes super-utilisateurs étranges.

PASSWORD_HISTORY Définit la longueur de l'historique des mots de passe pour empêcher leur réutilisation. Ceci n'est pas supporté par pam_tcb.

ENABLE_DNS_SPOOFING_PROTECTION Activer/Désactiver la protection contre l'usurpation d'adresse IP.

CHECK_SECURITY Permet des contrôles de sécurité quotidiens.

ALLOW_ROOT_LOGIN Autoriser/Forbattre la connexion directe à la racine.

CHECK_UNOWNED Permet de vérifier les fichiers non possédés.

ALLOW_USER_LIST Autoriser/envoyer la liste des utilisateurs du système sur les gestionnaires d'affichage (kdm et gdm).

NOTIFY_WARN Permet de prendre en charge les notifications de sécurité à l'aide de libnotify. Cela permet aux notifications de sécurité d'être envoyées directement sur le bureau de l'utilisateur.

ALLOW_REMOTE_ROOT_LOGIN Autoriser/Forbattre la connexion racine à distance via sshd. Vous pouvez spécifier oui, non et sans mot de passe. Voir la page de manuel sshd_config(5) pour plus d'informations.

ENABLE_MSEC_CRON Activer/Désactiver le contrôle de sécurité horaire msec.

ENABLE_SULOGIN Activer/Désactiver sulogin(8) au niveau utilisateur unique.

ALLOW_XAUTH_FROM_ROOT Permet/interdit d'exporter l'affichage lors du passage du compte root aux autres utilisateurs. Voir pam_xauth(8) pour plus de détails.

USER_UMASK Définissez le masque utilisateur umask.

ACCEPT_ICMP_ECHO Accepter/Refuser l'écho ICMP.

AUTHORIZE_SERVICES Configurez l'accès aux services tcp_wrappers (voir hosts.deny(5)).

Si arg = oui, tous les services sont autorisés. Si arg = local, seuls les services locaux le sont, et si arg = non, aucun service n'est autorisé. Dans ce cas, pour autoriser les services dont vous avez besoin, utilisez /etc/hosts.allow (voir hosts.allow(5)).

TTY_WARN Permet d'effectuer des contrôles de sécurité périodiques sur le terminal.