



Réseaux et Télécoms
IUT Nord Franche-Comté

**IUT Nord Franche-Comté
Montbéliard**
Département Réseaux et
Télécommunications
4 Place Lucien Tharradin
25200 Montbéliard



MS Solutions
15 Rue Auguste Jouchoux,
25000 Besançon

2023-2024

Mise en place d'une architecture réseau sécurisé pour entreprise

Rapport de fin d'études

LOUREIRO Hugo

Soutenue le 21/06/2024 à Montbéliard, devant le jury composé de :

LARTAUD Fabrice :
AOUBIZA Boujemaa :
GIVRON Stéphane :

Tuteur professionnel
Tuteur académique
Candide Jury

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à Monsieur Fabrice Lartaud de MS Solutions pour m'avoir accueilli chaleureusement dans son entreprise durant ces deux années d'alternance. Votre soutien et vos conseils m'ont permis de développer mes compétences professionnelles et de m'intégrer parfaitement au sein de votre entreprise.

Je souhaite également remercier l'IUT Réseau et Télécommunications de Montbéliard pour la qualité des enseignements tout au long de l'année. Les connaissances acquises ont été fondamentales pour mon développement personnel et professionnel. Les enseignants ont toujours été disponibles et à l'écoute, ce qui a grandement facilité mon apprentissage.

Un remerciement particulier à Monsieur Hakim Mabed pour ses cours sur la sécurité, notamment ceux portant sur pfsense. Ses explications claires et précises m'ont permis de mieux comprendre les enjeux de la sécurité informatique et d'acquérir des compétences pratiques essentielles dans ce domaine.

Cette année a été marquée par des rencontres enrichissantes, des défis stimulants et de nombreuses opportunités d'apprentissage. Chaque moment passé chez MS Solutions et à l'IUT de Montbéliard a contribué à mon épanouissement professionnel et personnel.

En conclusion, je suis profondément reconnaissant à tous ceux qui ont rendu cette expérience possible. Votre générosité et votre expertise ont eu un impact durable sur ma carrière. Merci infiniment.

RÉSUMÉ

Objectifs des projets :

Principe est de proposer différents services à moindre cout matériel donc tous c'est projet ont été créer sur PROXMOX (solution de virtualisations open source).

- Création d'une DMZ à trois niveaux via PFSense :

Concevoir et implémenter une zone démilitarisée (DMZ) à trois niveaux utilisant le pare-feu open-source PFSense. Cette DMZ doit isoler et protéger les ressources internes de l'entreprise tout en permettant un accès sécurisé depuis l'extérieur. Le projet vise à configurer des règles de pare-feu précises, segmenter le réseau efficacement et gérer les risques potentiels associés à l'accès externe.

- Mise en place d'un VPN Wireguard itinérant :

Implémenter un réseau privé virtuel (VPN) utilisant Wireguard pour les futurs clients de MS Solutions. Ce VPN doit offrir une connectivité sécurisée et fiable pour permettre aux employés d'accéder aux ressources de l'entreprise de manière sécurisée, peu importe leur localisation. Le projet doit garantir une simplicité d'utilisation, une rapidité de connexion et un haut niveau de sécurité.

- Création d'un logiciel de monitoring multi-site via lien VPN et remontée d'information par mail :

Développer un logiciel de surveillance multi-site capable de collecter et d'analyser des données à partir de différents sites connectés via VPN. Le logiciel doit inclure des fonctionnalités pour envoyer des alertes et des rapports par e-mail, assurant une surveillance proactive et continue des infrastructures réseau. Ce projet vise à améliorer la réactivité face aux incidents et à offrir une visibilité complète sur l'état des réseaux surveillés.

Table des matières

REMERCIEMENTS	0
RÉSUMÉ	2
1. Introduction	4
2. Présentation de MS Solutions	5
3. FONDEMENT THEORIQUE	6
4. LES PROJETS	7
4.1.1 CRÉATION DMZ A 3 NIVEAUX	7
4.1.2 Introduction avec Pfsense	8
4.1.3 Travail à faire	9
4.1.4 Conclusion	12
4.1.5 Problème rencontré	12
4.1.6 Évolution potentielle : Installation de Snort	13
4.2 Mise en place d'un VPN Wireguard itinérant	13
4.2.1 Mise en place d'un VPN Wireguard itinérant	14
4.2.2 Connaissance de base	15
4.2.3 Travail à faire	16
4.2.4 Explication des tests	22
4.2.5 Conclusion	23
4.2.6 Problème rencontré	23
4.2.7 Amélioration éventuel	24
4.3 Création d'un logiciel de monitoring multi-site	25
4.3.1 Travail à faire	26
4.3.2 Explication des tests	32
4.3.3 Conclusion	32
4.3.4 Problème rencontré	33
4.3.5 Amélioration éventuel	34
5. Bibliographie et webographie	35
6. Annexes	37

1. Introduction

Ce rapport de fin d'étude présente le fruit de mon expérience au sein de l'entreprise MS Solutions. MS Solutions est une entreprise dynamique et innovante qui se spécialise dans la fourniture de solutions informatiques aux petites et moyennes entreprises (PME). En tant qu'apprenti, j'ai eu l'opportunité d'explorer divers aspects des technologies réseau et de la cybersécurité, tout en contribuant aux projets clients et en acquérant une expérience pratique précieuse.

Mon alternance chez MS Solutions s'est articulée autour de trois principaux projets distincts, chacun visant à renforcer les infrastructures et les services offerts par l'entreprise à ses clients. Ces projets, détaillés dans ce rapport, sont les suivants :

- **Création d'une DMZ à trois niveaux via PFSense** : Ce projet avait pour objectif de concevoir et d'implémenter une zone démilitarisée (DMZ) à trois niveaux utilisant le pare-feu open-source PFSense. Ce travail a impliqué la configuration de règles de pare-feu, la segmentation du réseau et la gestion des risques potentiels.
- **Mise en place d'un VPN Wireguard itinérant** : Dans le cadre de ce deuxième projet, j'ai travaillé sur l'implémentation d'un réseau privé virtuel (VPN) utilisant Wireguard. Wireguard est réputé pour sa simplicité, sa rapidité et sa sécurité, et ce projet visait à fournir une solution VPN robuste pour permettre un accès sécurisé aux ressources de l'entreprise depuis n'importe où.
- **Création d'un logiciel de monitoring multi-site via lien VPN et remontée d'information par mail** : Le troisième projet consistait à développer un logiciel de surveillance multi-site capable de collecter et d'analyser des données à partir de différents sites connectés via VPN. Le logiciel inclut également des fonctionnalités pour envoyer des alertes et des rapports par e-mail, assurant ainsi une surveillance proactive et continue des infrastructures réseau.

Enfin, une conclusion récapitulera les principales leçons tirées de ces projets et proposera des pistes pour de futurs développements, reliant ainsi les trois parties de manière cohérente.

2. Présentation de MS Solutions

MS Solutions est société de services, fondée par Fabrice LARTAUD le 1er octobre 2015, est une entreprise spécialisée dans la fourniture de solutions informatiques complètes pour les petites et moyennes entreprises (PME). Depuis sa création, MS Solutions s'est imposée comme un acteur clé dans le domaine des technologies de l'information, offrant une gamme variée de services adaptés aux besoins spécifiques de ses clients.

L'entreprise compte parmi ses principaux clients des groupes industriels tels que IVECO (*voir Annexes 1*), avec des sites situés dans plusieurs départements français (70, 67, 68, 25, 21). En plus de ces grands groupes, MS Solutions collabore avec des entreprises locales du Doubs, incluant notamment des scieries, garage multi marque démontrant ainsi sa capacité à s'adapter aux exigences variées de différents secteurs d'activité.

Les services proposés par MS Solutions couvrent trois domaines principaux : la gestion de la téléphonie sur IP (VOIP), l'infrastructure et les réseaux, ainsi que la gestion de la sauvegarde et la mise en place de serveurs. Dans le domaine de la VOIP, l'entreprise offre des solutions de communication avancées qui permettent à ses clients de bénéficier d'une connectivité fiable et économique.

En ce qui concerne l'infrastructure et les réseaux, MS Solutions s'assure que les systèmes informatiques de ses clients sont robustes, sécurisés et capables de répondre aux défis actuels et futurs. MS solutions met en place des architectures réseau performantes, garantit une gestion efficace des données et assure une protection optimale contre les cybermenaces.

La gestion de la sauvegarde et la mise en place de serveurs constituent un autre pilier des services de MS Solutions. Consciente de l'importance cruciale des données pour les entreprises, l'entreprise propose des solutions de sauvegarde robustes et sécurisées. Elle accompagne également ses clients dans la configuration et la maintenance de serveurs, garantissant ainsi une disponibilité et une performance maximales des systèmes d'information.

Avec une approche axée sur la satisfaction client et une expertise technique reconnue, MS Solutions continue de se développer et d'innover pour répondre aux besoins croissants de ses clients. L'entreprise s'engage à fournir des solutions personnalisées et efficaces, contribuant ainsi à la réussite et à la pérennité des entreprises qu'elle accompagne.

3. FONDEMENT THEORIQUE

Les trois projets abordés dans ce rapport ont été initiés pour répondre aux attentes spécifiques de différents clients de MS Solutions, en cherchant à proposer des solutions efficaces et économiques. Chaque projet visait à résoudre des problèmes précis rencontrés par les clients, tout en s'appuyant sur les ressources disponibles au sein de mon entreprise.

Pour le premier projet, un client de MS Solutions souhaitait moderniser son infrastructure réseau en remplaçant son matériel obsolète. Le défi consistait à réaliser cette mise à jour avec un budget limité. Ma solution proposée impliquait la création d'une zone démilitarisée (DMZ) à trois niveaux en utilisant le pare-feu open-source PFsense. Cette DMZ permettrait de protéger les ressources internes tout en permettant un accès sécurisé depuis l'extérieur. En utilisant PFsense, une solution performante et économique, le projet visait à améliorer la sécurité du réseau du client sans dépasser le budget alloué.

Ensuite le deuxième projet, les commerciaux de IVECO, un des principaux clients de MS Solutions, avaient besoin d'un accès sécurisé aux ressources de l'entreprise lorsqu'ils étaient en déplacement. Traditionnellement, la configuration des VPN itinérants était complexe et fastidieuse. Le projet consistait à simplifier la création de ces VPN en mettant en place une solution basée sur Wireguard, connu pour sa simplicité et sa rapidité. En facilitant la configuration et l'utilisation des VPN, le projet visait à améliorer la productivité des commerciaux tout en garantissant la sécurité des communications.

Enfin pour un client ayant plusieurs entreprises, il était essentiel pour lui que son logiciel métier soit accessible dans tous ces entreprises via lien VPN. Cependant, aucune solution de surveillance n'existait pour monitorer ces connexions VPN. Le projet avait pour objectif de développer un logiciel de monitoring multisites qui non seulement collecterait et analyserait les données des différents sites, mais enverrait également des alertes et des rapports par e-mail en cas de problème. Cette solution permettrait une surveillance proactive et assurerait la continuité des opérations.

A savoir que les différents projets ont été créés dans le serveur labo de l'entreprise donc certaine limitation matérielle m'a obligé à m'adapter afin de proposer une solution viable.

4. LES PROJETS

4.1.1 CRÉATION DMZ A 3 NIVEAUX

Suite à une demande d'un client de l'amélioration de la sécurité réseau pour son réseau, j'ai entrepris la création d'une zone démilitarisée (DMZ) à trois niveaux en utilisant PFsense, hébergé sur un système d'exploitation Proxmox. Ce projet vise à offrir une solution sécurisée et économique pour isoler et protéger les ressources internes de l'entreprise tout en permettant un accès contrôlé depuis l'extérieur.

J'ai comme attention d'utiliser mes connaissances acquises lors de la SAE 503 qui avait pour but de créer une architecture sécurisée avec Proxmox lors de cette SAE j'ai proposé une architecture DMZ à 2 niveaux donc je compte bien l'améliorer.

Proxmox est une plateforme de virtualisation open-source qui permet de gérer des environnements virtualisés avec une grande flexibilité et efficacité. En utilisant Proxmox, je peux déployer et gérer des machines virtuelles et des conteneurs, offrant ainsi une solution robuste et économique pour la création de la DMZ. La flexibilité de Proxmox permet d'optimiser l'utilisation des ressources matérielles disponibles, réduisant ainsi les coûts tout en augmentant la sécurité et la fiabilité des systèmes.

Le projet de DMZ à trois niveaux implique la segmentation du réseau en trois zones distinctes, chacune contrôlée par des règles de pare-feu spécifiques via PFsense. La première zone est dédiée aux services accessibles depuis l'extérieur, la deuxième zone est destinée aux services internes de l'entreprise, et la troisième zone est réservée aux systèmes les plus sensibles nécessitant un niveau de sécurité maximal. Cette architecture permet de minimiser les risques de sécurité en isolant les différents niveaux de confiance et en appliquant des politiques de sécurité adaptées à chaque zone.

L'utilisation de Proxmox dans ce projet me permet de créer et de gérer facilement les différentes machines virtuelles nécessaires pour chaque niveau de la DMZ. Grâce à Proxmox, je peux également bénéficier de fonctionnalités avancées telles que la migration en direct, les sauvegardes automatisées et la gestion centralisée des ressources, assurant ainsi une administration simplifiée et une continuité de service optimale.

Ce projet répond à la demande du client de moderniser son infrastructure réseau avec un budget limité, tout en offrant une solution de sécurité robuste et évolutive. En

combinant les avantages de PFSense et de Proxmox, je suis en mesure de fournir une DMZ performante, sécurisée et adaptable aux besoins futurs de l'entreprise.

L'implémentation de cette DMZ à trois niveaux sur Proxmox représente une solution intégrée et économique, démontrant mon engagement à fournir des solutions innovantes et adaptées aux exigences de sécurité de nos clients.

4.1.2 Introduction avec PFSense

Dans ce projet, j'ai entrepris la création d'une DMZ (zone démilitarisée) à plusieurs niveaux en utilisant PFSense, une solution open-source de pare-feu et de routeur basée sur FreeBSD. PFSense est largement reconnue pour sa flexibilité, sa robustesse et ses nombreuses fonctionnalités. Contrairement à d'autres solutions commerciales comme Stormshield, PFSense offre une alternative économique sans compromettre les fonctionnalités ou la sécurité. Cette caractéristique en fait un choix idéal pour les établissements d'enseignement et les petites entreprises qui cherchent à maximiser leur rapport qualité-prix sans faire de compromis sur la sécurité réseau.

Avant le lancement de ce projet je suis d'abord renseignée sur PFSense afin de pouvoir mieux le comparer à c'est concurrent payant.

PFSense est un logiciel open-source qui transforme un ordinateur en un pare-feu puissant et flexible. Il est basé sur le système d'exploitation FreeBSD et est largement utilisé pour protéger les réseaux contre les menaces externes. Avec PFSense, on peut filtrer le trafic réseau pour empêcher les accès non autorisés et sécuriser les données de l'entreprise. Il est populaire parce qu'il est gratuit, facile à configurer et offre de nombreuses fonctionnalités avancées, comme la gestion des VPN (Réseau Privé Virtuel), qui permet des connexions sécurisées à distance.

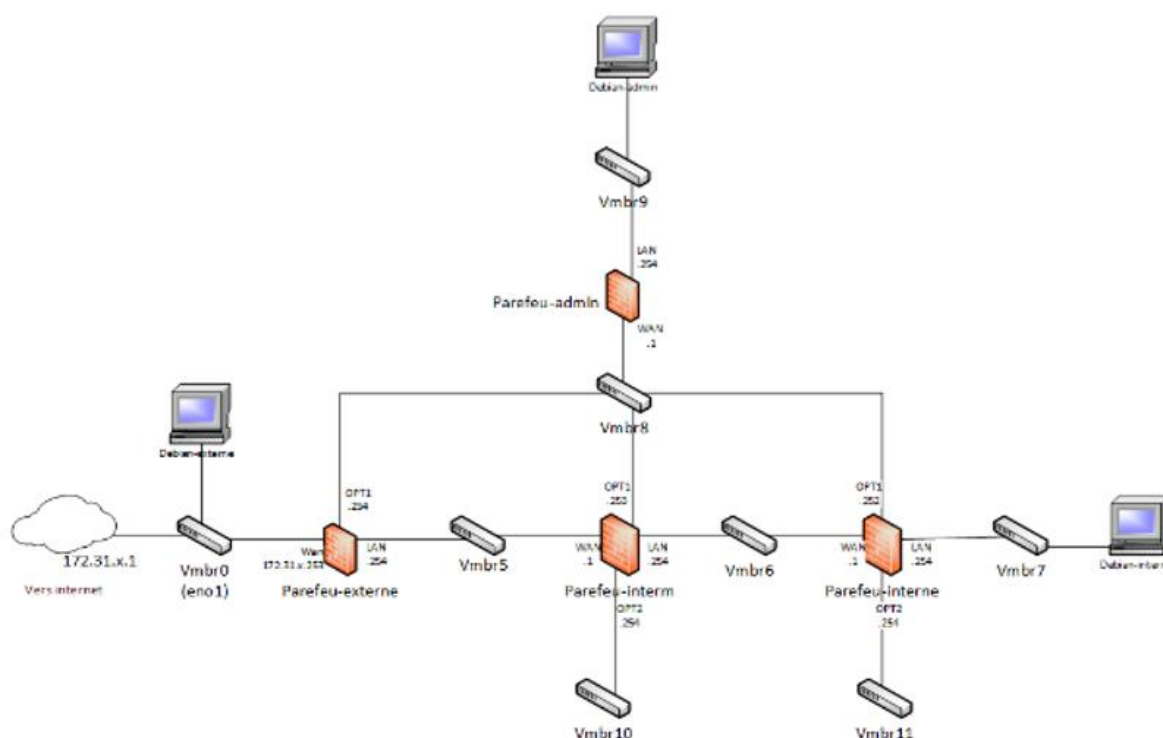
Dans un réseau, PFSense joue un rôle crucial en sécurisant l'accès aux ressources internes. Il contrôle quelles données peuvent entrer et sortir du réseau, protégeant ainsi contre les attaques de pirates informatiques et les logiciels malveillants. En plus de la sécurité, PFSense permet aussi de gérer et de prioriser le trafic réseau pour garantir que les applications importantes fonctionnent correctement. Par exemple, on peut s'assurer que les appels vidéo ou les applications de travail reçoivent toujours la bande passante nécessaire.

En résumé, PfSense est un outil essentiel pour toute organisation cherchant à sécuriser son réseau tout en maintenant une performance optimale. Il offre une solution économique et efficace pour gérer la sécurité et la connectivité du réseau, adaptée aussi bien aux petites entreprises qu'aux grandes organisations.

4.1.3 Travail à faire

Dans ce projet, je vais créer une DMZ à plusieurs niveaux. Je vais travailler avec un serveur Proxmox pour gérer nos machines virtuelles. L'objectif est de comprendre et de mettre en œuvre une architecture réseau sécurisée où différents segments du réseau sont isolés pour limiter les risques en cas de compromission. Voici un aperçu de la structure que je vais mettre en place :

Figure 1 : Schéma de l'architecture théorique du réseaux (MS solutions)



Plus de précision sur le schéma le but est d'affiner la sécurité entre chaque pare-feu voici en simplifiant le rôle de chacun :

Pare-feu externe : Il s'agit du premier point de défense contre les attaques venant de l'extérieur.

Pare-feu intermédiaire : Il contrôle le trafic entre le pare-feu externe donc les services exposés et le pare-feu interne les données sensibles de l'entreprise.

Pare-feu interne : Il protège les ressources sensibles et critiques au sein du réseau interne de l'entreprise.

Pare-feu administrateur : Il permet une gestion sécurisée des autres pare-feux et des composants réseau. Le principe est que c'est le seul des trois autres à ne pas avoir accès à internet pour plus de sécurité.

En utilisant Proxmox, je peux déployer et gérer des machines virtuelles et des conteneurs, offrant ainsi une solution robuste et économique pour la création de la DMZ. La flexibilité de Proxmox permet d'optimiser l'utilisation des ressources matérielles disponibles, réduisant ainsi les coûts tout en augmentant la sécurité et la fiabilité des systèmes.

Une des fonctionnalités clés de Proxmox est la gestion des réseaux virtuels, où le terme VMBR (Virtual Machine Bridge) joue un rôle central. Un VMBR est essentiellement un pont réseau virtuel qui permet aux machines virtuelles de communiquer entre elles et avec le réseau physique.

Le VMBR fonctionne en connectant les interfaces réseau des machines virtuelles à un pont virtuel, similaire à un switch réseau. Ce pont virtuel est relié à l'interface réseau physique de l'hôte Proxmox. Grâce à ce pont, les machines virtuelles peuvent envoyer et recevoir des paquets réseau, comme si elles étaient connectées directement à un switch physique. Cela permet de configurer des réseaux complexes, incluant la segmentation du réseau et l'isolement des machines virtuelles pour des raisons de sécurité.

En résumé, les VMBR sur Proxmox sont des outils puissants pour gérer les connexions réseau des machines virtuelles, offrant sécurité, flexibilité et performance. Ils sont essentiels pour créer des environnements de virtualisation robustes et sécurisés.

Il faudra ajouter de nouveaux VMBR (ponts virtuels) afin de connecter nos nouvelles machines virtuelles. Une fois les ponts configurés, je créerai de nouvelles machines virtuelles et attribuerai à chacune le VMBR approprié en respectant le schéma réseau établi. Cette étape est cruciale pour garantir que chaque machine virtuelle soit correctement isolée et puisse communiquer de manière sécurisée avec les autres composants de la DMZ.

Les manipulations PFsense ADMIN :

4.1.3 Travail à faire :

Pour commencer, je dois désactiver le RFC1918 sur l'interface WAN pour permettre les connexions à partir de l'internet public. Ensuite, je vérifierai qu'aucune route par défaut n'est utilisée pour éviter des chemins de communication non sécurisés. Enfin, j'activerai l'outbound NAT sur l'interface WAN pour permettre la traduction d'adresses réseau et sécuriser les communications sortantes.

Les manipulations PFsense autres :

En utilisant une machine virtuelle que je déplacerai entre les différents VMBR, je configurerai les autres pare-feux en suivant un ordre précis : PF interne, puis PF intermédiaire, et enfin PF externe. Ce séquençage est important pour assurer une configuration cohérente et éviter des conflits de configuration. Comme pour le pare-feu administrateur, je désactiverai le RFC1918 sur l'interface WAN et définirai une passerelle par défaut. Je créerai ensuite deux règles pour autoriser l'administration via OPT1 et désactiverai l'administration par le réseau LAN (Système > Advanced > Admin Access).

Le but à terme est de permettre uniquement au pare-feu admin de pouvoir modifier la configuration de chacun des pare-feux comme cela, il sera total impossible de les modifier via internet unique via un intranet interne au machine virtuel.

Un intranet est un réseau privé utilisé par une organisation, comme une entreprise ou une institution, pour partager des informations et des ressources de manière sécurisée parmi ses employés ou membres. Contrairement à Internet, qui est accessible à tout le monde, un intranet est limité aux utilisateurs autorisés au sein de l'organisation.

Accès web de puis le WAN :

Un serveur Apache2, souvent simplement appelé Apache, est un logiciel de serveur web open-source. Il est utilisé pour héberger des sites web et diffuser du contenu sur Internet. Apache est l'un des serveurs web les plus populaires et les plus utilisés dans le monde. Apache2 est un serveur web puissant et flexible qui joue un rôle crucial dans la diffusion de contenu web et le développement d'applications web. Sa popularité est due à sa simplicité d'utilisation, sa flexibilité et ses capacités de sécurité avancées.

J'ai installé un serveur web (apache 2) de test dans le réseau OPT2 du pare-feu intermédiaire. Une règle redirigera tout le trafic provenant de l'interface WAN vers le WAN du PF intermédiaire. J'appliquerai la même règle sur le pare-feu extérieur, permettant ainsi d'accéder au serveur via l'adresse IP 172.31.13.253. Cette configuration illustre comment rediriger et sécuriser le trafic réseau dans une architecture DMZ.

Le but est de faire comme si que l'entreprise aurait un site web donc qu'ils soient accessibles depuis internet

4.1.4 Conclusion

Ce projet de création d'une DMZ multi-niveaux et de gestion des pare-feux dans un environnement virtualisé est crucial pour comprendre la sécurité réseau. En réalisant cet exercice, j'ai acquis des compétences pratiques en configuration et gestion de la sécurité réseau, essentielles pour toute carrière en informatique.

L'utilisation de PfSense au lieu de Stormshield démontre un avantage notable pour ce projet. PfSense, en plus d'être gratuit, est open-source, ce qui lui permet d'avoir une large communauté d'utilisateurs et de développeurs qui contribuent régulièrement à son amélioration et à la résolution des problèmes. De plus, des ressources et des documentations abondantes sont disponibles en ligne. Stormshield, bien que disposant d'un support technique professionnel, peut ne pas offrir le même niveau d'interaction communautaire.

4.1.5 Problème rencontré

Lors de la création de la DMZ à trois niveaux, j'ai rencontré plusieurs défis significatifs qui ont nécessité des solutions créatives et une gestion rigoureuse. Le premier problème majeur était la limitation matérielle. Le serveur Proxmox sur lequel je travaillais disposait de peu de mémoire RAM, ce qui limitait la quantité de ressources que je pouvais allouer aux différentes machines virtuelles. Cette contrainte m'a obligé à optimiser chaque machine virtuelle de manière à ce qu'elles fonctionnent efficacement avec le minimum de ressources disponibles. J'ai dû ajuster les configurations et désactiver certains services non essentiels pour assurer le bon fonctionnement de l'ensemble de l'infrastructure.

Un autre défi important a été la gestion de mon temps. En plus de ce projet, je continuais à gérer mes responsabilités professionnelles habituelles, qui incluaient les appels clients et les dépannages sur site. La combinaison de ces tâches quotidiennes avec les exigences techniques et temporelles de la création de la DMZ a rendu la gestion du temps cruciale. J'ai dû planifier soigneusement mes journées, équilibrant les urgences clients avec les phases critiques du projet. Cela a impliqué des sessions de travail prolongées et une priorisation stricte des tâches pour garantir que le projet avance tout en maintenant un niveau de service élevé pour les clients. Malgré ces

défis, la rigueur dans la gestion du temps et l'optimisation des ressources m'ont permis de mener à bien ce projet complexe.

4.1.6 Évolution potentielle : Installation de Snort

Snort est un système de détection d'intrusion réseau (IDS) open-source développé par Sourcefire, maintenant une filiale de Cisco. Utilisé pour surveiller le trafic réseau en temps réel, Snort analyse les paquets de données pour détecter des activités suspectes ou malveillantes. Il fonctionne en capturant les paquets circulant sur le réseau et en les comparant à une base de données de signatures d'attaques connues. Lorsqu'une correspondance est trouvée, Snort génère une alerte qui peut être utilisée pour prendre des mesures préventives immédiates ou pour enquêter plus en profondeur sur l'incident. L'une des forces de Snort est sa flexibilité : il peut être configuré non seulement pour détecter des intrusions, mais aussi pour prévenir les attaques en bloquant automatiquement les paquets malveillants.

L'ajout de Snort présente plusieurs avantages : il permet une détection rapide des intrusions en identifiant une variété d'attaques et de menaces telles que les tentatives d'exploitation de vulnérabilités et les scans de port, permettant une réponse rapide aux comportements anormaux. Snort envoie des alertes instantanées lorsqu'une activité suspecte est détectée, ce qui permet aux administrateurs réseau d'intervenir immédiatement pour protéger les systèmes. En surveillant et enregistrant le trafic réseau, Snort fournit des informations précieuses sur les types de trafic et aide à identifier des tendances pouvant indiquer des problèmes de sécurité. De plus, en complément de PFSense, Snort ajoute une couche de protection supplémentaire, rendant le système plus résistant aux attaques en bloquant automatiquement les adresses IP malveillantes.

4.2 Mise en place d'un VPN Wireguard itinérant

Après avoir mis en place une DMZ à trois niveaux en utilisant PFSense et Proxmox, j'ai pu sécuriser efficacement le réseau interne de l'entreprise, isolant ainsi les différentes zones de confiance pour minimiser les risques de compromission.

Cependant, la sécurité réseau ne se limite pas à la protection des ressources internes. Avec l'augmentation du télétravail et des déplacements professionnels, il est devenu essentiel de garantir un accès sécurisé aux ressources de l'entreprise pour les utilisateurs distants. C'est dans ce contexte que le deuxième projet prend toute son importance : la mise en place d'un VPN Wireguard itinérant.

L'objectif de ce projet est de permettre aux employés de se connecter de manière sécurisée au réseau de l'entreprise depuis n'importe où dans le monde. Wireguard, choisi pour sa simplicité de configuration et ses performances accrues, se présente comme une solution idéale par rapport à d'autres options comme OpenVPN ou IPSec. Ce projet vise à offrir une connectivité sécurisée, rapide et fiable, tout en assurant une gestion simplifiée tant pour les administrateurs que pour les utilisateurs finaux.

La mise en place d'un VPN Wireguard implique plusieurs étapes cruciales, allant de l'installation et la configuration du serveur sur Debian 11, à la configuration des clients Wireguard pour les utilisateurs distants. En outre, il est important de tester la performance et la sécurité du VPN pour s'assurer qu'il répond aux besoins de l'entreprise. Ce projet, tout comme celui de la DMZ, nécessite une gestion rigoureuse des ressources et du temps, ainsi qu'une attention particulière aux détails pour garantir une solution efficace et fiable.

Avec cette transition vers le déploiement de Wireguard, je vais explorer comment offrir une sécurité de bout en bout pour les utilisateurs distants, complétant ainsi les mesures de sécurité internes mises en place avec la DMZ. L'objectif final est de créer une infrastructure réseau sécurisée et flexible, capable de répondre aux besoins actuels et futurs de l'entreprise.

4.2.1 Mise en place d'un VPN Wireguard itinérant

Un réseau privé virtuel (VPN) est une technologie qui permet de créer une connexion sécurisée entre un utilisateur et un réseau privé à travers Internet. En utilisant un VPN, les données sont cryptées, garantissant que toute information transmise reste confidentielle et protégée contre les interceptions. Les VPN sont largement utilisés pour diverses raisons, notamment pour accéder à des ressources réseau de manière sécurisée depuis des emplacements distants, protéger la vie privée des utilisateurs en masquant leur adresse IP et contourner les restrictions géographiques pour accéder à du contenu en ligne.

L'un des éléments clés de nombreux VPN modernes est l'utilisation du protocole UDP (User Datagram Protocol). Contrairement à TCP (Transmission Control Protocol), UDP est un protocole de communication qui ne nécessite pas d'établissement de connexion et ne garantit pas la livraison des paquets. Cette absence de mécanismes de contrôle de flux et de retransmission rend UDP plus rapide et plus efficace pour certaines applications, comme les jeux en ligne, les appels VoIP et les vidéos en streaming, où la vitesse est plus critique que la fiabilité absolue des paquets.

WireGuard, par exemple, est un protocole VPN moderne qui utilise UDP pour ses transmissions. En exploitant les avantages de l'UDP, WireGuard offre des connexions VPN rapides et performantes avec une latence réduite. Cette rapidité est essentielle pour les utilisateurs qui nécessitent des performances élevées et une connexion fluide, même lors de l'accès à des ressources distantes. De plus, la simplicité de la configuration de WireGuard, associée à ses algorithmes de cryptographie modernes, en fait une solution de choix pour les entreprises et les particuliers souhaitant sécuriser leurs communications sans sacrifier la performance.

Ainsi, un VPN utilisant UDP combine les bénéfices de la sécurité et de la performance, fournissant une solution optimale pour des environnements nécessitant des connexions rapides et sécurisées. Que ce soit pour accéder à des fichiers d'entreprise, naviguer sur Internet en toute sécurité, ou maintenir une communication fluide pour des applications en temps réel, l'utilisation d'un VPN basé sur UDP représente une avancée significative dans le domaine des technologies de réseau.

4.2.2 Connaissance de base

Principe des clés privée et publique :

Le principe des clés privée et publique est essentiel dans la cryptographie asymétrique, utilisée pour sécuriser les communications sur Internet. Chaque utilisateur possède une paire de clés : une clé privée, qui reste secrète, et une clé publique, qui est partagée librement. Lorsqu'Alice veut envoyer un message sécurisé à Bob, elle utilise la clé publique de Bob pour chiffrer le message. Seule la clé privée de Bob peut déchiffrer ce message, garantissant que seul Bob peut lire son contenu.

De même, Bob peut signer numériquement un message avec sa clé privée pour prouver son identité. Alice peut utiliser la clé publique de Bob pour vérifier cette signature, assurant ainsi que le message provient bien de Bob et qu'il n'a pas été altéré.

Ce système offre une sécurité robuste sans nécessiter le partage de clés secrètes, réduisant les risques de compromission. La clé publique peut être diffusée largement sans compromettre la sécurité, tandis que la clé privée reste confidentielle. Ce principe est largement utilisé dans les protocoles HTTPS pour sécuriser les sites web, les VPN pour des connexions sécurisées, et les signatures numériques pour garantir l'authenticité des documents électroniques.

Ainsi, le système de clés privée et publique permet de sécuriser efficacement les communications et les transactions numériques, assurant confidentialité, intégrité et authenticité.

Principe des Protocol de communication :

UDP (User Datagram Protocol) utilisé par WireGuard :

- **Performance :** UDP est plus rapide car il n'y a pas de contrôle de flux, d'accusé de réception ou de retransmission des paquets perdus. Cela réduit la latence et permet une transmission plus rapide des données, ce qui est essentiel pour des applications nécessitant une communication en temps réel, comme les jeux en ligne ou les appels VoIP.
- **Flexibilité :** En n'exigeant pas de connexion établie, UDP permet une plus grande flexibilité pour les connexions VPN. Les clients peuvent se reconnecter plus facilement après une interruption sans avoir à établir une nouvelle session.
- **Moins de surcharge de réseau :** Sans les mécanismes de correction d'erreur de TCP, UDP impose une moindre surcharge sur le réseau, ce qui améliore encore les performances globales.

TCP (Transmission Control Protocol) utilisé par OpenVPN :

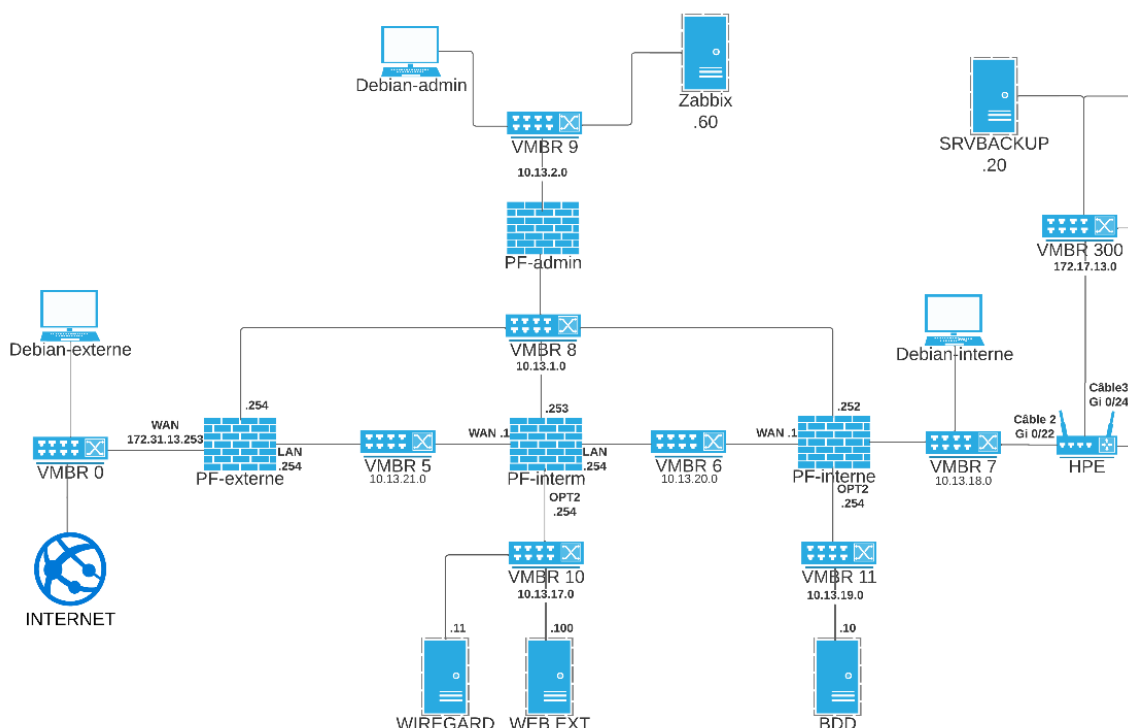
- **Fiabilité :** TCP offre une transmission fiable avec des mécanismes intégrés pour assurer que les données arrivent correctement. En cas de perte de paquets, TCP les retransmet, ce qui garantit l'intégrité des données.
- **Contrôle de flux et gestion de congestion :** TCP gère le contrôle de flux et la congestion, ce qui permet de maintenir la stabilité du réseau en ajustant dynamiquement la vitesse de transmission des données selon la capacité du réseau.

4.2.3 Travail à faire

Dans ce projet, nous allons créer un VPN pour permettre à un utilisateur distant d'accéder à ses fichiers depuis chez lui. Pour cela, nous mettrons en place un serveur VPN WireGuard fonctionnant sous Debian 11. L'objectif est de garantir un accès sécurisé et fiable aux ressources de l'entreprise, même à distance. Le choix de WireGuard s'explique par sa simplicité de configuration et ses performances accrues par rapport à d'autres solutions VPN comme OpenVPN ou IPsec. En outre, WireGuard est connu pour sa faible empreinte mémoire et son efficacité énergétique, ce qui le rend particulièrement adapté aux environnements nécessitant des

performances optimales avec des ressources limitées. Grâce à ces avantages, nous pouvons assurer une expérience utilisateur fluide et sécurisée, essentielle dans le contexte actuel de télétravail croissant. Voici un aperçu de la structure que je vais mettre en place :

Figure 2 : Schéma repensant le réseau actuellement (MS Solutions)



Installation et Configuration du serveur WireGuard :

Pour commencer, il nous faut une Debian 11 installée et paramétrée. Debian est choisie pour sa stabilité et sa large adoption dans le monde professionnel, ce qui assure une documentation abondante et un support communautaire solide.

`apt-get update`

Installer WireGuard Serveur:

`apt-get install Wireguard`

WireGuard est préféré pour sa légèreté et son efficacité. Il utilise des algorithmes de cryptographie modernes et bien optimisés, garantissant ainsi des connexions rapides et sécurisées.

Générer les clés privées et publiques est une étape essentielle pour la sécurisation des communications. Les clés sont générées de la manière suivante :

`wg genkey | sudo tee /etc/wireguard/wg-private.key | wg pubkey | sudo tee /etc/wireguard/wg-public.key`

Les clés privées doivent rester confidentielles et les clés publiques seront partagées avec les clients pour établir la connexion.

4.2.3 Travail à faire :

```
sudo cat /etc/wireguard/wg-private.key
```

Créer le fichier de configuration wg0.conf :

```
sudo nano /etc/wireguard/wg0.conf
```

Ajouter le contenu suivant :

```
[Interface]
```

```
Address = 10.7.0.1
```

```
SaveConfig = true
```

```
ListenPort = 51820
```

```
PrivateKey = <clé privée du serveur>
```

Cette configuration définit l'adresse IP du serveur VPN, le port d'écoute et la clé privée du serveur. Le choix du port par défaut (51820) peut être modifié pour des raisons de sécurité, en évitant les ports bien connus et potentiellement ciblés par des attaques.

Démarrer l'interface :

```
sudo wg-quick up wg0
```

Vérifier la configuration :

```
Ip a
```

Ces commandes permettent de vérifier que l'interface VPN est correctement configurée et active. La commande « wg show » fournit des détails sur l'état de l'interface et les pairs connectés.

Activer l'interface wg0 au démarrage pour garantir que le VPN est toujours disponible après un redémarrage du serveur :

```
sudo systemctl enable wg-quick@wg0.service
```

Modifier le fichier sysctl.conf pour activer le routage IP, ce qui permet à la machine de router les paquets entre les réseaux :

```
sudo nano /etc/sysctl.conf
```

Ajouter à la fin du fichier :

```
net.ipv4.ip_forward = 1
```

L'activation du routage IP est essentielle pour permettre au serveur VPN de transmettre les paquets entre les différents réseaux.

Installer « ufw », un pare-feu simple à utiliser mais puissant :

```
apt install ufw
```

4.2.3 Travail à faire :

Autoriser les ports nécessaires pour le SSH(22) et WireGuard (51820):

```
sudo ufw allow 22/tcp
```

```
sudo ufw allow 51820/udp
```

Grace a cette info peut s'en servir pour modifier le fichier before.rules pour ajouter les règles de NAT :

```
nano /etc/ufw/before.rules
```

Ajouter les lignes suivantes :

```
# NAT - IP masquerade
```

```
*nat
```

```
:POSTROUTING ACCEPT [0:0]
```

```
-A POSTROUTING -o ens18 -j MASQUERADE
```

```
# End each table with the 'COMMIT' line or these rules won't be processed
```

```
COMMIT
```

Et toujours dans le même fichier on va déclarer le réseau interne de l'entreprise (je vais l'adapter pour seulement un hôte mais cela est pareil pour un réseau il faut juste adapter le /32 en /24 ou autre). Ces règles permettent de masquer l'adresse IP du serveur et d'autoriser le routage des paquets entre les différents sous-réseaux, assurant ainsi une communication fluide et sécurisée :

```
# autoriser le forwarding pour le réseau distant de confiance (+ le réseau du VPN)
```

```
-A ufw-before-forward -s 172.17.13.1/32 -j ACCEPT
```

```
-A ufw-before-forward -d 172.17.13.1/32 -j ACCEPT
```

```
-A ufw-before-forward -s 10.13.17.11/32 -j ACCEPT
```

```
-A ufw-before-forward -d 10.13.17.11/32 -j ACCEPT
```

```
-A ufw-before-forward -s 10.7.0.2/32 -j ACCEPT
```

```
-A ufw-before-forward -d 10.7.0.2/32 -j ACCEPT
```

Bon il ne reste plus qu'à appliquer nos changements et à redémarrer les services avec les commandes suivantes :

```
sudo ufw enable
```

```
sudo systemctl restart ufw
```

Configuration du client Wireguard :

La configuration initiale d'un client WireGuard peut sembler complexe pour un nouvel utilisateur, nécessitant la génération et l'échange de clés. Cependant, une fois cette étape franchie, l'utilisation et la gestion des connexions VPN deviennent extrêmement simples et rapides, offrant une sécurité robuste avec un minimum d'effort.

Télécharger et installer WireGuard depuis le site officiel. Le choix de WireGuard pour le client s'explique par sa compatibilité avec divers systèmes d'exploitation et sa facilité de configuration.

Nous allons créer un nouveau tunnel vide dans l'application WireGuard et ajouter un tunnel vide :

Nous allons créer un peer un serveur à distance ce qu'il faut savoir c'est que le logiciel wireguard va chercher de bloc un qui se nomme « [interface] » et l'autre « [Peer] » les crochets sont très importants donc je vais donner les deux blocs et vous aurez juste à remplacer avec vos valeurs cela sera plus simple.

[Interface]

```
PrivateKey = OP6dceP4+C5QwSFXg0uXcQ2PiLG9gJpgTW1Hte+4q2s=
```

```
Address = 10.7.0.2/24
```

```
DNS = 8.8.8.8
```

Cette configuration permet de définir l'adresse IP du client, le serveur DNS à utiliser, ainsi que les adresses IP et sous-réseaux accessibles via le VPN.

[Peer]

```
PublicKey = byF3zSV9rMrglgR3PFaOFAq6G8SpSit+9k7ePb9y8B8=
```

```
AllowedIPs = 10.7.0.2/24, 10.13.17.11/32, 172.17.13.1/32
```

```
Endpoint = 172.31.13.253:51820
```

Alors un peu d'explication pour la configuration du « Peer »

-PublicKey : il s'agit de la clé publique du serveur WireGuard Debian 11 (vous pouvez obtenir sa valeur via la commande "sudo wg")

-AllowedIPs : il s'agit des adresses IP / des sous-réseaux accessibles via ce réseau VPN WireGuard, ici il s'agit du sous-réseau propre à mon VPN WireGuard (10.7.0.2/24) et de mon LAN distant (172.17.13.1/32)

-Endpoint : L'endpoint correspond à l'adresse publique du serveur WireGuard.

Bon il ne reste plus que à déclarer le nouveau client wireguard est tout sera bon :

On va d'abord finir avec le client wireguard il reste plus que a déclarer le peer sur le serveur WireGuard il faut d'abord stopper l'interface wg 0 :

```
sudo wg-quick down /etc/wireguard/wg0.conf
```

Ensuite modifier le fichier wg0.conf pour ajouter les informations du client :

```
nano /etc/wireguard/wg0.conf
```

Voici les informations qui faut rajouter dans le fichier à la suite du bloc « [interface] » on va rajouter cela :

```
[Peer]
```

```
PublicKey = PbkwKFpLaqXINQWu7ycaWz0dRsA3OCyu4j5p6EGNTA=
```

```
AllowedIPs = 10.7.0.2/32
```

Ce bloc contient la clé publique du client ainsi que l'adresse IP allouée à celui-ci.

Il ne reste plus qu'à sauvegarder le fichier et a relancez l'interface « wg0 » :

```
wg-quick up /etc/wireguard/wg0.conf
```

Pour finir la config, on va limiter l'accès aux fichiers de configuration pour garantir que seules les personnes autorisées peuvent les modifier :

```
sudo chmod 600 /etc/wireguard/ -R
```

Cette command exprime que le fichier ne pourra plus que être lu par une personne identifier sur la machine. Il ne nous reste que a effectuer les tests.

4.2.4 Explication des tests

Après avoir mis en place le VPN WireGuard et effectué des tests de performance (*voir Annexes 2*) (*voir Annexes 3*), j'ai comparé ses résultats avec ceux obtenus avec OpenVPN. WireGuard s'est distingué par sa simplicité de configuration et son efficacité. Lors des tests de débit, WireGuard a montré une latence significativement plus faible par rapport à OpenVPN. En utilisant le protocole UDP, WireGuard optimise la transmission des données en réduisant les délais associés aux mécanismes de contrôle de flux et de retransmission, typiques de TCP utilisé par OpenVPN. Cela se traduit par une vitesse de connexion plus rapide et une meilleure performance globale, particulièrement bénéfique pour des applications nécessitant une communication en temps réel comme les appels VoIP et les jeux en ligne.

En termes de stabilité, WireGuard a également surpassé OpenVPN. Les connexions établies avec WireGuard ont démontré une robustesse remarquable, avec moins de perturbations et des reconnections plus fluides après des interruptions. Cette fiabilité est essentielle pour les utilisateurs mobiles qui dépendent d'une connexion VPN stable pour accéder aux ressources de l'entreprise de manière sécurisée.

Un autre aspect crucial est la consommation de ressources. WireGuard, grâce à son design minimaliste et ses algorithmes de cryptographie modernes, utilise moins de CPU et de mémoire comparé à OpenVPN. Cette efficacité permet de déployer WireGuard sur des matériels avec des ressources limitées, sans compromettre la sécurité ou la performance.

Les tests de sécurité ont montré que WireGuard offre une sécurité robuste, comparable à OpenVPN, mais avec une configuration plus simple. Les clés cryptographiques utilisées par WireGuard sont générées et gérées plus facilement, réduisant ainsi le risque d'erreurs humaines. De plus, la simplicité des configurations de WireGuard contribue à une maintenance plus facile et à une gestion des clés plus sécurisée.

En résumé, les tests ont clairement indiqué que WireGuard surpasse OpenVPN en termes de performance, stabilité et efficacité. En intégrant WireGuard dans notre

infrastructure, nous bénéficions d'un VPN moderne qui est non seulement performant et facile à gérer, mais aussi capable de fournir une sécurité robuste. Les utilisateurs peuvent ainsi profiter d'une connexion VPN rapide et fiable, assurant une expérience utilisateur optimale et une protection renforcée des données.

4.2.5 Conclusion

La mise en place d'un VPN itinérant avec WireGuard sous Debian 11 présente plusieurs avantages significatifs par rapport à OpenVPN, principalement en termes de simplicité, de performance et de sécurité.

WireGuard est réputé pour sa configuration simple, rapide mais aussi sécurisé. Contrairement à OpenVPN, qui nécessite de gérer de nombreux paramètres et fichiers de configuration, WireGuard se base sur des principes minimalistes. La configuration se fait principalement via quelques commandes et des fichiers de configuration légers, ce qui facilite grandement le déploiement et la gestion quotidienne, de plus l'utilisation du protocole UDP permet une nette performance d'in point vu débit par rapport à OpenVPN en outre le système de cryptographie utiliser et complexe a craqué afin d'apporter une sécurité notable.

En résumé, WireGuard, avec son utilisation d'UDP, offre des avantages clairs en termes de simplicité, de performance et de flexibilité, en particulier pour les applications nécessitant une faible latence et des débits élevés. OpenVPN, utilisant TCP, peut offrir une fiabilité accrue. Le choix entre les deux dépend donc des besoins spécifiques et des contraintes du réseau de l'utilisateur.

En intégrant WireGuard dans notre infrastructure, nous bénéficions d'un VPN moderne, performant et facile à gérer, tout en assurant une sécurité robuste et une expérience utilisateur optimale pour nos utilisateurs.

4.2.6 Problème rencontré

Lors de l'installation du VPN WireGuard, plusieurs défis ont émergé, similaires à ceux rencontrés dans le projet de création de la DMZ à trois niveaux, notamment la gestion du temps. En effet, jongler entre mes responsabilités professionnelles quotidiennes, comme les appels clients et les dépannages sur site, et les exigences techniques de ce projet, a nécessité une planification minutieuse et une gestion rigoureuse du temps.

Un autre problème majeur a été la mise en service du VPN. Contrairement à certains autres VPN, WireGuard ne fournit pas de messages d'erreur explicites lorsque la

4.2.7 Amélioration éventuel :

liaison ne parvient pas à se connecter. Cette absence de retour d'information clair a rendu le dépannage plus complexe. Identifier l'origine exacte des erreurs de connexion a parfois été laborieux, nécessitant de vérifier manuellement chaque configuration et chaque étape du processus.

Cette difficulté a été exacerbée par la nécessité de s'assurer que les clés cryptographiques étaient correctement générées et échangées, et que les règles de pare-feu et de routage étaient correctement appliquées. Ces obstacles ont exigé une attention particulière aux détails et une patience considérable pour résoudre les problèmes et assurer la fonctionnalité du VPN WireGuard.

Malgré ces défis, surmonter ces difficultés a renforcé mes compétences en gestion de projet et en résolution de problèmes techniques.

4.2.7 Amélioration éventuel

Une évolution éventuelle pour le VPN itinérant serait de le transformer en VPN WireGuard site à site.

Cette configuration permettrait de créer une connexion permanente entre deux réseaux distincts, facilitant l'interconnexion sécurisée des ressources et des services de chaque site. En passant à un VPN site à site, l'entreprise bénéficierait de plusieurs avantages significatifs.

Premièrement, cela simplifierait l'accès aux ressources partagées entre les différents bureaux ou sites, éliminant le besoin pour chaque utilisateur de se connecter individuellement au VPN.

Deuxièmement, cette approche améliorerait la gestion centralisée du réseau, permettant une administration plus efficace et cohérente des règles de sécurité et des configurations réseau.

Troisièmement, un VPN site à site pourrait optimiser la performance globale du réseau en réduisant la latence et en augmentant la vitesse de transfert de données grâce à une connexion directe entre les sites.

En outre, cela renforcerait la continuité des opérations, assurant une communication ininterrompue et sécurisée entre les sites, ce qui est crucial pour les activités qui nécessitent une disponibilité constante des services.

Enfin, cette configuration offrirait une robustesse accrue contre les pannes de connexion individuelles, puisque le tunnel VPN entre les sites resterait actif même si certains utilisateurs rencontrent des problèmes de connexion. En somme, évoluer vers un VPN WireGuard site à site pourrait grandement améliorer l'efficacité, la sécurité et la résilience du réseau de l'entreprise.

4.3 Création d'un logiciel de monitoring multi-site

Après avoir mis en place un VPN itinérant avec WireGuard, garantissant une connexion sécurisée et fiable pour les utilisateurs distants, il est crucial de penser à la surveillance et à la gestion de l'ensemble de l'infrastructure réseau. La sécurité et la connectivité sont des aspects fondamentaux, mais sans une surveillance adéquate, il devient difficile de garantir la disponibilité continue et la performance optimale des services. C'est ici qu'intervient la nécessité d'un logiciel de monitoring efficace. Un bon système de surveillance permet de détecter rapidement les problèmes potentiels, de suivre les performances en temps réel et de prendre des mesures proactives pour éviter les interruptions de service.

Pour répondre à ces besoins, le prochain projet se concentrera sur la mise en place d'un logiciel de dashboard en utilisant Dashy, combiné avec Uptime Kuma pour le monitoring. Dashy est un outil open-source qui permet de créer des tableaux de bord interactifs et personnalisables. Il offre une interface conviviale pour visualiser divers indicateurs de performance et l'état de différents services. Uptime Kuma, de son côté, est un logiciel de surveillance qui permet de suivre la disponibilité et le temps de réponse des services en ligne. En intégrant ces deux outils, nous pouvons créer une solution complète de monitoring qui non seulement surveille la disponibilité des services mais présente également ces informations de manière claire et accessible.

Le but principal du logiciel de monitoring que nous mettrons en place avec Dashy et Uptime Kuma est de vérifier la stabilité des liens VPN (*voir Annexes 4*) via des pings réguliers. En effectuant des pings sur les différents liens VPN, le système de monitoring peut évaluer en temps réel la disponibilité et la qualité des connexions entre les sites. Cette vérification continue est essentielle pour détecter rapidement toute instabilité ou interruption de service.

Chaque lien VPN sera surveillé par des pings automatiques, et les résultats seront affichés sur le tableau de bord de Dashy. Les indicateurs de performance montreront le temps de réponse et le taux de disponibilité de chaque connexion. En cas de perte de connexion ou de temps de réponse anormalement élevé, des alertes seront générées pour permettre une intervention rapide.

Cette fonctionnalité de monitoring apporte plusieurs avantages.

4.3.1 Travail à faire :

Tout d'abord, elle permet de garantir que les connexions VPN restent stables et performantes, ce qui est crucial pour les communications et les transferts de données entre les sites de l'entreprise.

Ensuite, elle aide à identifier et à diagnostiquer rapidement les problèmes de réseau, réduisant ainsi le temps d'arrêt et minimisant l'impact sur les opérations de l'entreprise.

Enfin, en fournissant une visibilité claire et accessible sur l'état des liens VPN, le tableau de bord aide les administrateurs à prendre des décisions informées pour l'optimisation et la maintenance du réseau.

En intégrant cette capacité de surveillance dans notre infrastructure, nous assurons une gestion proactive et efficace des connexions VPN, garantissant une continuité de service et une performance optimale pour les utilisateurs. Cette approche renforce la résilience et la fiabilité du réseau, contribuant ainsi à la sécurité et à l'efficacité globales de l'infrastructure informatique de l'entreprise.

4.3.1 Travail à faire

Dans ce projet, nous allons mettre en place un tableau de bord interactif et personnalisable en utilisant Dashy, fonctionnant dans un conteneur Docker(*voir Annexes 5*), pour surveiller et vérifier la stabilité des liens VPN. Docker est une technologie qui permet de créer et de gérer des conteneurs, qui sont des environnements isolés dans lesquels des applications peuvent s'exécuter. Les conteneurs incluent tout ce dont une application a besoin pour fonctionner, comme les bibliothèques, les dépendances et le code, garantissant que l'application fonctionne de manière cohérente sur n'importe quel système.

L'utilisation de Docker pour héberger Dashy présente plusieurs avantages. Docker simplifie le déploiement et la gestion des applications en isolant les environnements de développement et de production, réduisant ainsi les problèmes de compatibilité. En utilisant Dashy dans un conteneur Docker, nous pouvons facilement déployer, mettre à jour et gérer le tableau de bord sans affecter les autres services ou applications sur le serveur.

L'objectif est de créer un système de surveillance efficace qui vérifie la disponibilité et la qualité des connexions VPN via des pings réguliers, assurant ainsi la continuité et la performance des services réseau. Le choix de Dashy s'explique par sa flexibilité et sa convivialité, permettant de créer des tableaux de bord personnalisés adaptés aux besoins spécifiques de l'entreprise. De plus, nous intégrerons Uptime Kuma pour fournir des données détaillées sur les temps de réponse et la disponibilité des liens

4.3.1 Travail à faire :

VPN, ce qui est crucial pour détecter rapidement toute instabilité ou interruption de service.

En utilisant Dashy et Uptime Kuma, nous pouvons surveiller efficacement les liens VPN, assurant une communication fluide et sécurisée entre les sites de l'entreprise. Cette solution de monitoring est essentielle pour garantir une performance optimale des services réseau et pour réagir rapidement en cas de problème. Grâce à cette approche, nous pouvons offrir une expérience utilisateur robuste et fiable, répondant aux exigences de l'infrastructure moderne.

Installation et Configuration du serveur WireGuard :

Pour commencer, il nous faut une Debian 11 installée et paramétrée. Debian est choisie pour sa stabilité et sa large adoption dans le monde professionnel, ce qui assure une documentation abondante et un support communautaire solide.

`apt-get update`

Docker nécessite certains paquets pour fonctionner correctement. Installez-les en exécutant la commande suivante :

`apt-get install apt-transport-https ca-certificates curl gnupg lsb-release`

Installez Docker Engine en exécutant la commande suivante :

`sudo apt-get install docker-ce docker-ce-cli containerd.io`

Utilisez la commande suivante pour créer et exécuter le conteneur Docker pour Dashy :

```
docker run -d \  
-p 8080:80 \  
-v ~/dashy/config.yml:/app/public/conf.yml \  
--name dashy \  
--restart=always \  
lissy93/dashy:latest
```

Explication des options :

4.3.1 Travail à faire :

-d : Démarre le doker en arrière-plan .

-p 80:80 : Mappe le port 80 du conteneur au port 80 de l'hôte. Vous pouvez accéder à Dashy via <http://localhost:80>.

-v ~/dashy/config.yml:/app/public/conf.yml : Monte le fichier de configuration personnalisé (facultatif) dans le conteneur.

--name dashy : Donne un nom au conteneur.

--restart=always : Assure que le conteneur redémarre automatiquement en cas de panne ou de redémarrage de l'hôte.

lissy93/dashy:latest : Utilise l'image Docker officielle de Dashy.

Après avoir exécuté cette commande, vous pouvez accéder à Dashy via votre navigateur web en vous rendant à l'adresse <http://localhost:80>

Bon il nous reste encore à paramètre l'interface de gestion global nous allons donc éditer le fichier de configuration de dashy

[Nano dashy/config.yml](#)

Nano sert simplement à éditer un fichier text

Voici une partie du fichier de configuration de celui que j'ai créé pour mon logiciel :

- name: MSSOLUTIONS

widgets:

- type: stat-ping

useProxy: true

updateInterval: 20

options:

hostname: <http://192.168.28.66:8080>

showChart: false

showInfo: false

frameHeight: 350

id: 0_880_statping

displayData:

4.3.1 Travail à faire :

```

sortBy: default
rows: 1
cols: 1
collapsed: false
hideForGuests: falseS
filteredItems:
  - &ref_6
    title: LOGs
    url: http://192.168.28.66:3001/dashboard
    id: 0_880_logs
items:
  - *ref_6

```

Alors je vais vous expliquer que veut dire chaque ligne afin que vous compreniez ce que j'ai voulue crée (rappel : ceci n'est que un secteur du document final) :

Nom de la section : MSSOLUTIONS

Widgets :

Type : stat-ping - Un widget de type "stat-ping" est utilisé pour surveiller l'état d'un service en effectuant des pings à intervalles réguliers.

Utiliser Proxy : true - Indique que le widget utilise un proxy pour les requêtes.

Intervalle de Mise à Jour : 20 secondes - Le widget effectue des mises à jour toutes les 20 secondes.

Options :

Nom d'hôte : http://192.168.28.66:8080 - L'URL de l'hôte à surveiller.

Afficher le Graphique : false - Le graphique n'est pas affiché.

Afficher les Informations : false - Les informations supplémentaires ne sont pas affichées.

Hauteur du Cadre : 350 pixels - La hauteur du cadre du widget.

Identifiant : 0_880_statping - Identifiant unique pour ce widget.

Données d'Affichage :

4.3.1 Travail à faire :

Trier Par : default - Critère de tri par défaut.

Lignes : 1 - Nombre de lignes pour l'affichage.

Colonnes : 1 - Nombre de colonnes pour l'affichage.

Réduit : false - Indique que la section n'est pas réduite.

Cacher pour les Invités : false - La section n'est pas cachée pour les invités.

Items Filtrés :

Titre : LOGs - Le titre de l'item filtré.

URL : <http://192.168.28.66:3001/dashboard> - L'URL du service des logs.

Identifiant : 0_880_logs - Identifiant unique pour cet item.

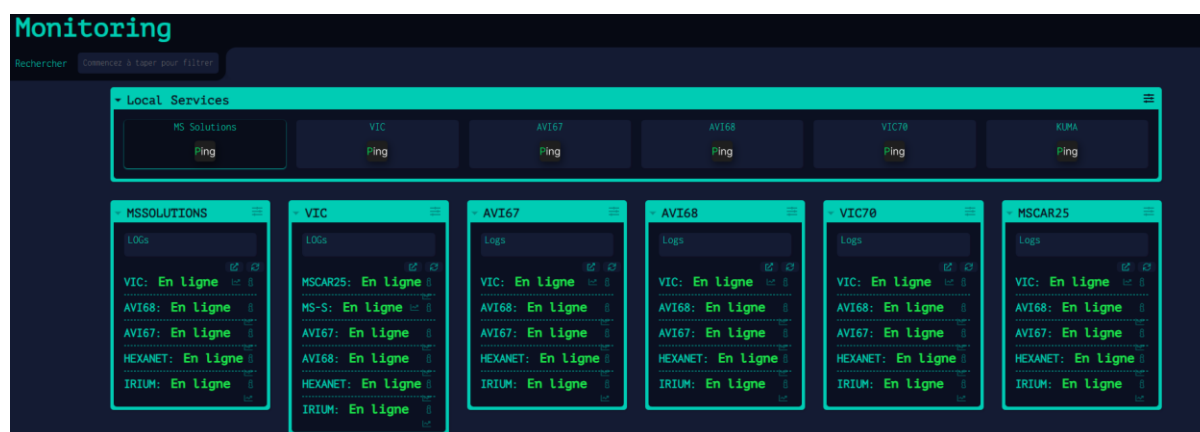
Items :

Référence : *ref_6 - Référence à l'item filtré précédemment défini.

Après avoir créé le secteur MS Solutions j'en ai créé un à chaque site où je voulais vérifier l'état de connexion des différents VPN en adaptant bien sûr les IP à celle des réseaux attendus.

Une fois le document fini le rendu final et pas trop mal :

Figure 3 : page web du logiciel fini (MS solutions)



Une fois que le dossier de configuration a été finalisé, j'ai procédé à l'installation du logiciel sur chacun des différents serveurs. Cette installation a permis de configurer une interface de monitoring centralisée, essentielle pour la surveillance proactive de notre infrastructure. Chaque serveur a été équipé du logiciel, ce qui permet d'effectuer des pings réguliers sur les différents sites que nous gérons. Grâce à cette configuration, nous sommes maintenant capables de vérifier l'état de chaque service en temps réel.

L'interface de monitoring comprend des widgets de type "stat-ping" qui envoient des requêtes ping à intervalles réguliers aux serveurs spécifiés. En cas de défaillance ou de latence inhabituelle, ces widgets fournissent des alertes instantanées, nous permettant d'identifier rapidement les problèmes potentiels avant qu'ils n'affectent gravement nos opérations. De plus, chaque section de l'interface comprend un tableau de bord des logs, fournissant des détails précis sur chaque événement enregistré.

Cette configuration permet non seulement de surveiller l'état des services, mais aussi d'anticiper et de gérer efficacement les incidents. Ainsi, lorsque des anomalies sont détectées, nous pouvons accéder directement aux logs détaillés pour diagnostiquer et résoudre les problèmes. Cette surveillance continue et détaillée assure que notre infrastructure reste robuste et fiable, minimisant les temps d'arrêt et garantissant la continuité des services pour nos utilisateurs et clients.

Une fois cela fait il ne me restait plus qu'à lier un Uptime Kuma afin que l'on puisse être avertie à chaque problème :

Uptime Kuma est un outil essentiel pour garantir la disponibilité continue de nos services. En intégrant Uptime Kuma dans notre infrastructure, nous avons mis en place un système de surveillance qui nous permet de recevoir des alertes par email en temps réel en cas de déconnexion d'un lien VPN. Grâce à Uptime Kuma, nous pouvons surveiller activement l'état de nos connexions VPN, et ainsi, être immédiatement informés de toute interruption de service.

L'utilisation de ce logiciel se traduit par une surveillance en continu des liens VPN critiques. Lorsque Uptime Kuma détecte une déconnexion, une alerte est automatiquement envoyée à notre équipe via email. Cette fonctionnalité est cruciale car elle nous permet de réagir rapidement pour résoudre les problèmes de connexion avant qu'ils n'affectent les utilisateurs finaux. De plus, les notifications par email incluent des détails spécifiques sur l'incident, facilitant ainsi le diagnostic et la résolution rapide.

En résumé, Uptime Kuma renforce notre capacité à maintenir des connexions VPN stables et fiables. Les alertes en temps réel assurent que nous sommes toujours au courant des interruptions, nous permettant d'intervenir immédiatement et de minimiser les temps d'arrêt. Ce niveau de réactivité est indispensable pour garantir la continuité des services et maintenir la satisfaction de nos utilisateurs et clients.

Un fois Uptime Kuma activer avec dashy il ma fallu rentré les codes pour me connecter à une boîte mail (il faut bien qu'ils puissent envoyer des mail) et voilà tout était fait à chaque problème de VPN un mail partait nous avertir se qui donne un avantage considérable sur la gestion d'incident.

4.3.2 Explication des tests

Le test effectué avec Uptime Kuma, démontrant la puissance et la flexibilité de cet outil de surveillance(*voir Annexes 6*). Le test montre une notification envoyée via un bot Discord lorsque le lien "Test - Domain" est détecté comme étant en panne, et une autre notification lorsque le lien est à nouveau opérationnel. Cette fonctionnalité critique permet de surveiller la disponibilité des services en temps réel et d'être immédiatement informé de tout incident.

Uptime Kuma est extrêmement puissant grâce à sa capacité à s'adapter aux besoins spécifiques des clients en matière de notifications. Que ce soit par Discord, Telegram, ou par email, Uptime Kuma offre une flexibilité totale pour configurer les alertes selon les préférences de l'utilisateur. Les notifications peuvent inclure des informations détaillées telles que le nom du lien, l'URL, le timestamp, le statut de l'erreur, et les temps de ping, ce qui permet une réactivité rapide et efficace en cas de problème.

Cette flexibilité de configuration des notifications assure que les équipes peuvent être averties par leur canal de communication préféré, garantissant ainsi une réactivité immédiate. Cela est essentiel pour maintenir la continuité des services, minimiser les temps d'arrêt, et assurer une expérience utilisateur optimale. En résumé, Uptime Kuma est un outil indispensable pour toute organisation cherchant à surveiller et maintenir la disponibilité de ses services avec des notifications personnalisées et efficaces.

4.3.3 Conclusion

En conclusion, ce projet a démontré l'efficacité et la robustesse de notre solution de surveillance des services en utilisant des outils comme Uptime Kuma et une configuration personnalisée via YAML. En installant et configurant méticuleusement le logiciel sur nos différents serveurs, nous avons pu établir un système de monitoring centralisé capable de vérifier l'état des services en temps réel, d'effectuer des pings réguliers, et de fournir des alertes instantanées en cas de défaillance. La mise en place d'une interface de logs a également renforcé notre capacité à diagnostiquer rapidement les problèmes et à prendre des mesures correctives immédiates.

L'intégration d'Uptime Kuma, avec sa flexibilité de notifications par Discord, Telegram, ou email, a prouvé sa valeur en nous permettant de rester informés en temps réel des déconnexions de liens VPN et autres incidents critiques. Cette approche proactive nous assure une réactivité optimale, minimisant les interruptions de service et garantissant la continuité des opérations. La personnalisation des alertes selon les préférences des utilisateurs améliore encore notre capacité à intervenir rapidement.

Ce projet illustre l'importance d'un système de surveillance bien conçu pour maintenir la disponibilité et la fiabilité des services, ainsi que l'efficacité des outils modernes de monitoring pour anticiper et gérer les incidents. En somme, cette expérience a renforcé notre infrastructure, amélioré notre réactivité face aux problèmes, et contribué à une meilleure satisfaction de nos utilisateurs et clients.

4.3.4 Problème rencontré

Lors de l'intégration de Dashy et Uptime Kuma, plusieurs défis ont émergé, similaires à ceux rencontrés dans le projet de création de la DMZ à trois niveaux, notamment la gestion du temps. En effet, jongler entre mes responsabilités professionnelles quotidiennes, comme les appels clients et les dépannages sur site, et les exigences techniques de ce projet, a nécessité une planification minutieuse et une gestion rigoureuse du temps.

Un autre problème majeur a été la mise en service des outils de surveillance. Contrairement à certaines autres solutions, la configuration initiale de Dashy et Uptime Kuma ne fournit pas toujours des messages d'erreur explicites lorsque la connexion ou les pings ne parviennent pas à fonctionner correctement. Cette absence de retour d'information clair a rendu le dépannage plus complexe. Identifier l'origine exacte des erreurs a parfois été laborieux, nécessitant de vérifier manuellement chaque configuration et chaque étape du processus.

Cette difficulté a été corrigée par la nécessité de s'assurer que les paramètres de notification étaient correctement configurés et que les règles de pare-feu et de routage étaient correctement appliquées pour permettre une communication fluide entre les services. Ces obstacles ont exigé une attention particulière aux détails et une patience considérable pour résoudre les problèmes et assurer la fonctionnalité optimale de Dashy et Uptime Kuma.

Malgré ces défis, surmonter ces difficultés a renforcé mes compétences en gestion de projet et en résolution de problèmes techniques. La mise en place réussie de ces outils

de surveillance a non seulement amélioré la réactivité de notre système aux incidents, mais a également démontré la robustesse et la flexibilité des solutions Dashy et Uptime Kuma dans un environnement complexe.

4.3.5 Amélioration éventuel

Une évolution éventuelle pour notre système de surveillance utilisant Dashy et Uptime Kuma serait de transformer notre configuration actuelle en une solution plus intégrée et automatisée. En utilisant ces outils de manière synergique, nous pourrions créer une infrastructure de monitoring plus robuste et réactive, facilitant la gestion et la sécurité des services.

Dashy offre une interface de tableau de bord hautement personnalisable, permettant une visualisation claire et centralisée des statuts de nos services. En intégrant Uptime Kuma, nous ajoutons la capacité de recevoir des alertes en temps réel via divers canaux comme Discord, Telegram, ou par email, nous assurant ainsi une réactivité immédiate aux incidents. Cette configuration permettrait une surveillance continue et une gestion efficace des services.

Premièrement, cette intégration simplifierait l'accès aux informations critiques sur la santé des services, éliminant le besoin pour chaque utilisateur de vérifier individuellement chaque service. Deuxièmement, l'approche combinée améliorerait la gestion centralisée des notifications et des alertes, permettant une administration plus cohérente et efficace des règles de sécurité et des configurations de monitoring.

Troisièmement, l'utilisation conjointe de Dashy et Uptime Kuma pourrait optimiser la performance globale du réseau de surveillance en réduisant les délais de réaction et en augmentant la précision des informations grâce à des mises à jour en temps réel. En outre, cette combinaison renforcerait la continuité des opérations, assurant une surveillance ininterrompue et sécurisée des services critiques, ce qui est crucial pour les activités nécessitant une disponibilité constante.

Enfin, cette configuration offrirait une robustesse accrue contre les pannes de service individuelles, puisque le système de monitoring resterait opérationnel même si certains éléments du réseau rencontrent des problèmes. En somme, évoluer vers une intégration complète de Dashy et Uptime Kuma pourrait grandement améliorer l'efficacité, la sécurité et la résilience de notre système de surveillance, garantissant une gestion optimale et une réactivité exemplaire face aux incidents.

5. Bibliographie et webographie

IVECO AFC - Concessionnaires Alsace et Franche-Comté. (2024b, mai 31). *IVECO AFC - concessionnaires Alsace et Franche-Comté*. <https://ivecoafc.fr/>

PfSense® - world's most trusted open source firewall. (s. d.). <https://pfsense.com/>

Donenfeld, J. A. (s. d.). WireGuard : fast, modern, secure VPN tunnel. <https://www.wireguard.com/>

Docker : Accelerated Container Application Development. (2024, 20 mai). Docker. <https://www.docker.com/>

Dashy | Dashy. (s. d.). <https://dashy.to/>

Uptime kuma. (s. d.). <https://uptime.kuma.pet/>

Agence nationale de la sécurité des systèmes d'information. (s. d.). <https://cyber.gouv.fr/>

6. Abstract

LOUREIRO Hugo

15 Rue Auguste Jouchoux, 25000 Besançon

LARTAUD Fabrice

2022-2024

Setting up a roaming Wireguard VPN

Virtual Private Network (VPN) / encryption / nomad vpn

WireGuard is a modern and efficient VPN solution known for its simplicity, performance, and security. Unlike traditional VPNs like OpenVPN, which use TCP, WireGuard operates over UDP. This makes it faster and more suitable for real-time applications such as online gaming and VoIP. Setting up WireGuard involves installing it on a Debian 11 server, generating cryptographic keys for secure communication, and configuring both the server and clients.

WireGuard's minimalist design focuses on ease of use, with straightforward configuration files and commands. The server configuration includes setting up IP addresses, private keys, and port numbers. On the client side, you need to install WireGuard and create a tunnel with the appropriate server settings.

The performance benefits of WireGuard are significant. Its use of modern cryptographic algorithms ensures secure and rapid data transmission. Additionally, its lightweight protocol reduces latency, making it ideal for environments where speed and reliability are crucial.

In summary, WireGuard offers a streamlined, high-performance alternative to traditional VPNs. Its simplicity in configuration, combined with robust security features, makes it an excellent choice for both individual and enterprise use, ensuring secure and efficient remote access to resources.

7. Annexes

Annexes 1.....	38
Annexes 2.....	38
Annexes 3.....	38
Annexes 4.....	39
Annexes 5.....	39
Annexes 6.....	40

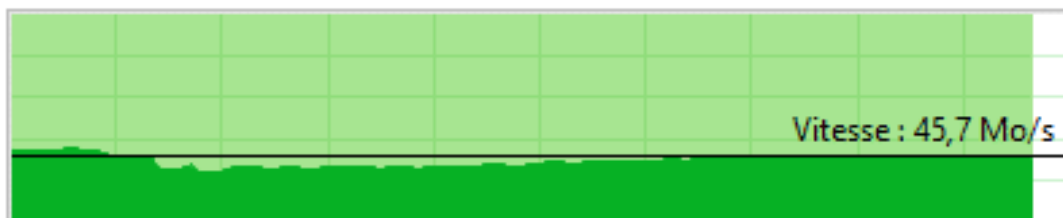
Annexes 1 : Logo IVECO groupe



Annexes 2 : Test débit Wireguard et OpenVPN

Copie d'un élément de Téléchargements vers Partage

94% terminé



Nom : VMware-ESXi-7.0U3b-18905247-depot.zip

Temps restant : Environ 5 secondes

Éléments restants : 1 (20,5 Mo)

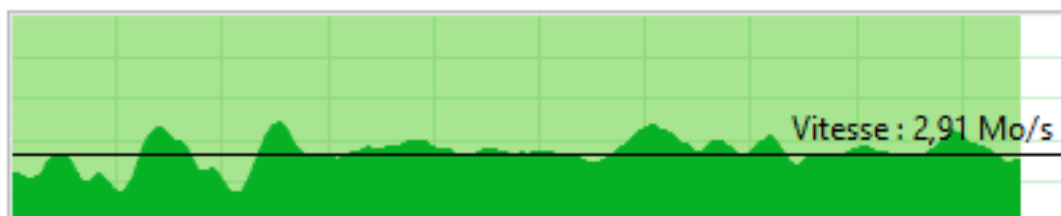
WireGuard VPN

^ Moins de détails

Annexes 3 : Test débit OpenVPN

Copie d'un élément de Téléchargements vers Partage

95% terminé



Nom : VMware-ESXi-7.0U3b-18905247-depot.zip

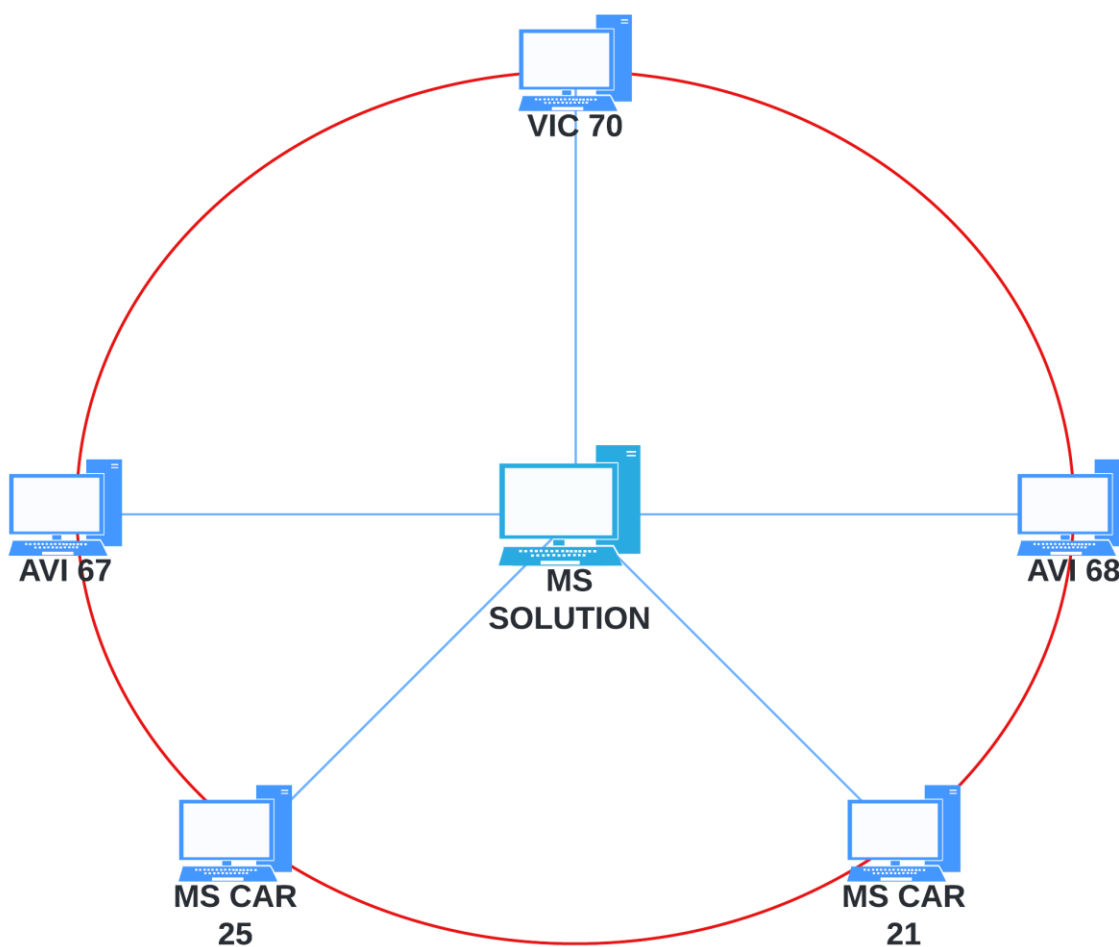
Temps restant : Environ 10 secondes

Éléments restants : 1 (16,5 Mo)

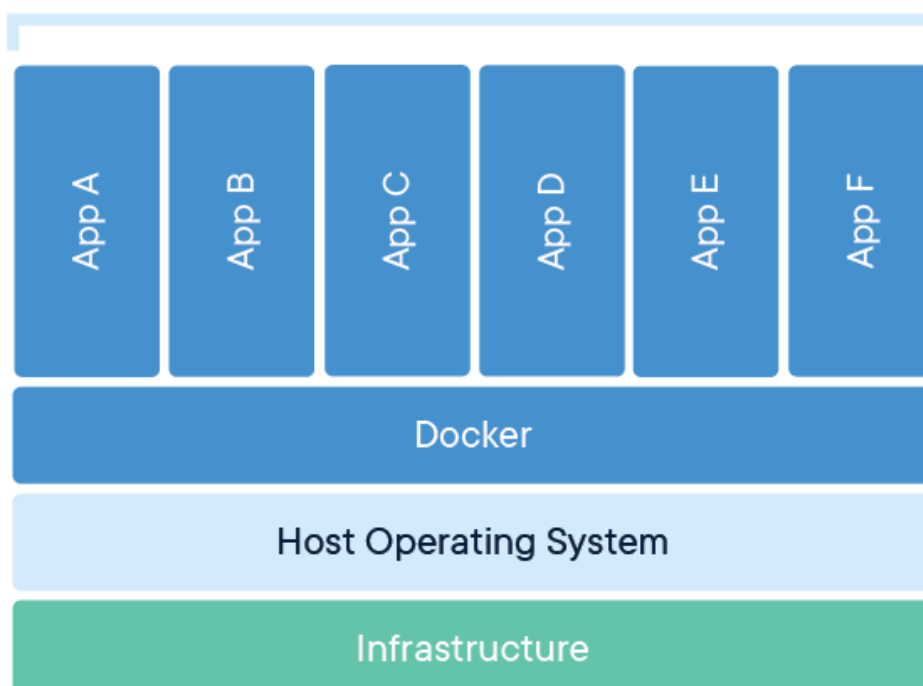
OpenVPN

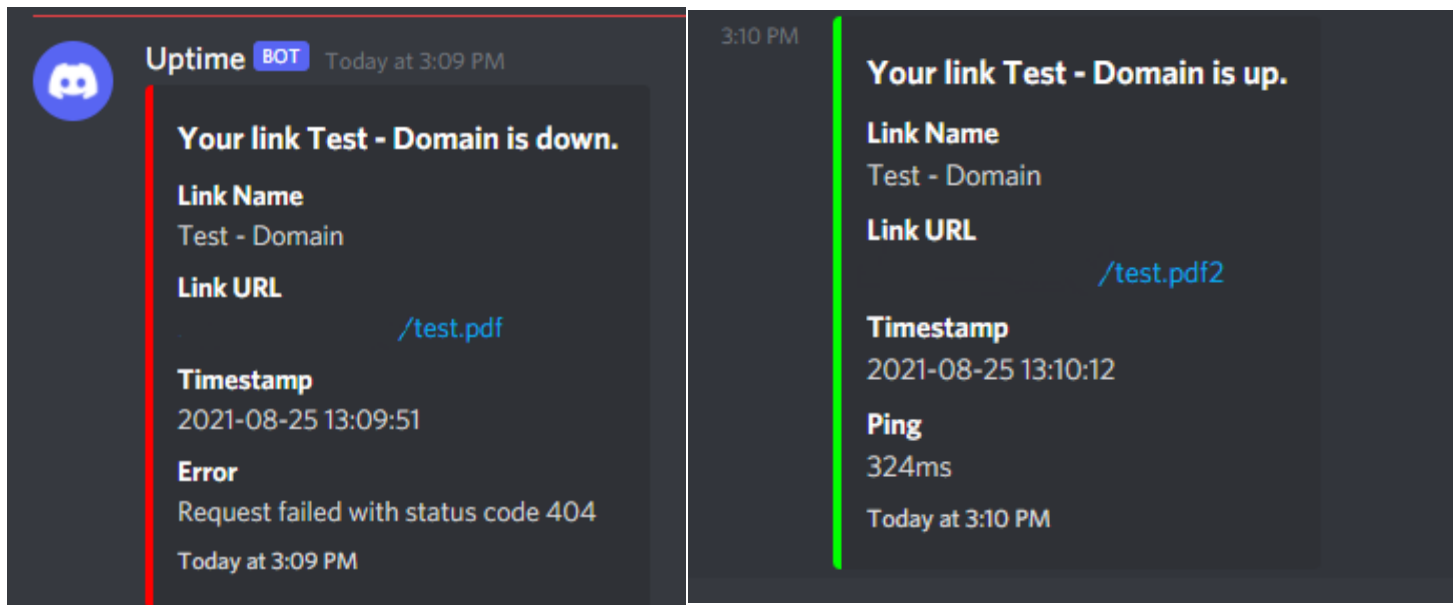
^ Moins de détails

Annexes 4 : Schéma Lien VPN Client IVECO :



Annexes 5 : Schéma d'un docker
Containerized Applications



Annexes 6 : Exemple de message de déconnection ou reconnections de lien VPN

The image displays two screenshots of Discord messages from the Uptime BOT. The left screenshot shows a message at 3:09 PM stating 'Your link Test - Domain is down.' with details: Link Name 'Test - Domain', Link URL '/test.pdf', Timestamp '2021-08-25 13:09:51', and Error 'Request failed with status code 404'. The right screenshot shows a message at 3:10 PM stating 'Your link Test - Domain is up.' with details: Link Name 'Test - Domain', Link URL '/test.pdf2', Timestamp '2021-08-25 13:10:12', and Ping '324ms'.

Uptime BOT Today at 3:09 PM

Your link Test - Domain is down.

Link Name
Test - Domain

Link URL
/test.pdf

Timestamp
2021-08-25 13:09:51

Error
Request failed with status code 404

Today at 3:09 PM

3:10 PM

Your link Test - Domain is up.

Link Name
Test - Domain

Link URL
/test.pdf2

Timestamp
2021-08-25 13:10:12

Ping
324ms

Today at 3:10 PM