MISE EN PLACE D'UN SERVEUR APACHE GUACAMOLE

Cédric BAUDIMENT
AgroMousquetaires – Site d'Antartic 2

BTS SIO SISR
Greta Drôme / Ardèche
2024 / 2026

I/ Cahier des charges

1- Descriptif de l'existant

L'existant repose sur un poste informatique fixe utilisé comme plateforme de virtualisation. Ce poste est équipé du logiciel VMware Workstation, permettant l'exécution de plusieurs machines virtuelles au sein d'un même environnement matériel. Il dispose également d'une connexion à Internet, utilisée pour les tests d'accès distant.

L'infrastructure mise en place est entièrement virtualisée et s'inscrit dans un environnement de test. Elle a pour objectif de simuler une architecture réseau sécurisée, comparable à celle rencontrée en entreprise, sans impacter un système d'information en production.

Cette infrastructure comprend un pare-feu pfSense, chargé d'assurer la sécurité du réseau interne, ainsi que plusieurs serveurs virtualisés sous Linux et Windows, destinés à être accessibles à distance. Un poste client sous Windows 10 est utilisé pour tester les connexions à distance depuis le réseau interne ou externe.

L'environnement existant a pour finalité de tester et valider des mécanismes d'accès distant sécurisés dans un cadre pédagogique, en s'appuyant sur des technologies couramment utilisées en milieu professionnel.

2- Besoins

Dans le cadre de cet environnement de test, plusieurs besoins ont été identifiés afin de permettre un accès distant fonctionnel et sécurisé aux ressources internes.

Il est nécessaire de disposer d'une solution d'accès distant centralisée, permettant à un utilisateur de se connecter aux serveurs internes depuis un poste client, sans accès direct au réseau local.

Le besoin principal est de garantir un accès distant sécurisé aux serveurs Linux et Windows, en limitant l'exposition des services sensibles (SSH, RDP) vers l'extérieur. Les connexions doivent être contrôlées et protégées par des mécanismes d'authentification.

L'infrastructure doit également permettre une gestion simplifiée des accès, avec une administration centralisée des connexions et des utilisateurs. Les accès doivent être limités aux seules ressources nécessaires.

Enfin, la solution mise en place doit être adaptée à un environnement de test, facilement déployable, reproductible et sans impact sur un système d'information de production, tout en restant représentative d'un usage professionnel.

3- Contraintes

La mise en place de l'infrastructure d'accès distant s'effectue dans un environnement de test entièrement virtualisé, ce qui implique des contraintes liées aux ressources matérielles du poste hôte. Les performances dépendent directement de la capacité du PC fixe à supporter plusieurs machines virtuelles simultanément.

L'infrastructure repose sur l'utilisation de VMware Workstation, ce qui limite le déploiement aux fonctionnalités offertes par cet outil de virtualisation et à la configuration des réseaux virtuels disponibles.

La solution d'accès distant doit être compatible avec des systèmes hétérogènes, notamment des serveurs Linux et un serveur Windows Server 2022, tout en restant accessible depuis un poste client sous Windows 10.

Des contraintes de sécurité doivent également être respectées. Les accès distants doivent être contrôlés, et l'exposition des services internes vers l'extérieur doit être limitée afin de ne pas fragiliser l'environnement de test.

Enfin, le projet s'inscrit dans un cadre pédagogique, ce qui impose une solution compréhensible, documentable et reproductible, sans recours à des services externes ou à des infrastructures complexes.

II/ Ressources

1- Ressources mises à disposition

Plusieurs ressources matérielles et logicielles sont mises à disposition pour la réalisation de ce TP.

Un poste informatique fixe est utilisé comme support principal de l'infrastructure. Il dispose des ressources matérielles nécessaires à l'exécution de plusieurs machines virtuelles simultanément ainsi que d'une connexion à Internet, indispensable pour l'installation des systèmes et les tests d'accès distant.

Le logiciel de virtualisation VMware Workstation est utilisé pour la création et la gestion des machines virtuelles. Il permet de configurer des réseaux virtuels isolés et de simuler une infrastructure réseau complète.

Les images ISO nécessaires au déploiement des machines virtuelles sont également disponibles :

- une image ISO de pfSense pour le pare-feu
- une image ISO de Debian 13, utilisée pour l'installation du serveur Apache Guacamole ainsi que des serveurs Linux cibles
- une image ISO de Windows Server 2022 pour le serveur Windows
- une image ISO de Windows 10 pour le poste client de test.

2- Ressource nécessaire à la mise en place

Pour déployer l'infrastructure de test et préparer l'expérimentation de différentes solutions d'accès distant, il est nécessaire de prévoir des ressources matérielles et logicielles adaptées à chaque machine virtuelle.

Chaque VM doit être configurée avec les ressources minimales nécessaires à son fonctionnement :

- La VM pfSense dispose d'un espace disque de 5 Go, de 256 Mo de mémoire et d'un cœur CPU.
- Les deux VM Debian 13, utilisées pour les serveurs Linux et le futur serveur d'accès distant, sont configurées avec 20 Go de disque, 2 Go de mémoire et un cœur CPU chacune.
- La VM Windows Server 2022 est allouée 40 Go de disque, 4 Go de mémoire et 2 cœurs CPU.
- Le poste client Windows 10 dispose également de 40 Go de disque, 4 Go de mémoire et 2 cœurs CPU.

En plus de ces ressources, les images ISO nécessaires à l'installation de chaque système doivent être disponibles, et un accès administrateur est requis sur chaque machine pour l'installation et la configuration.

3- Gestion des ressources

L'ensemble du projet a été déployé sur un poste physique unique hébergeant cinq machines virtuelles.

- Le serveur Debian 13 héberge Apache Guacamole, centralisant les accès distants vers les serveurs Linux et Windows Server 2022.
- Une seconde VM Debian 13 constitue le serveur Linux cible.
- La VM Windows Server 2022 sert de serveur Windows cible.
- Le poste client Windows 10 permet de tester les connexions à distance via l'interface web de Guacamole.
- Une VM pfSense joue le rôle de pare-feu et de passerelle entre le réseau interne (LAN) et l'accès Internet (WAN), simulant un environnement réseau sécurisé et réaliste.

Cette configuration assure un routage correct entre les VM, permet de tester les mécanismes d'accès distant dans des conditions proches d'un réseau d'entreprise et facilite la reproduction du projet. Les configurations réseau, règles du pare-feu et paramètres de Guacamole ont été documentés afin de simplifier la maintenance et le dépannage.

III/ Analyse

1- Descriptifs des solutions

a) Connexion directe RDP / SSH depuis le poste client

Cette solution consiste à accéder directement à un serveur ou poste distant depuis un autre poste en utilisant les protocoles natifs : RDP (Remote Desktop Protocol) pour les systèmes Windows et SSH (Secure Shell) pour les systèmes Linux. Le poste client établit la connexion directement vers la machine cible, sans serveur intermédiaire ni interface centralisée.

- Principe de fonctionnement : le client utilise l'outil natif du système pour se connecter au poste ou serveur distant. L'utilisateur saisit les identifiants d'accès et se retrouve face à l'environnement distant comme s'il était physiquement présent.
- Avantages :
    - Accès direct aux machines sans installation de logiciel supplémentaire (pour RDP et SSH natifs).
    - Configuration minimale sur le client et la machine cible.
    - Faible latence, connexion rapide et réactive.
- Points à considérer :
    - Chaque machine distante doit être accessible depuis le poste client, ce qui peut nécessiter l'ouverture de ports sur le réseau ou sur un pare-feu.
    - La gestion des utilisateurs et des sessions se fait individuellement sur chaque machine.
    - Moins adaptée pour un grand nombre de postes ou pour centraliser les accès, car il n'y a pas de gestion centralisée des connexions.

Cette solution reste simple et efficace pour la prise en main à distance ponctuelle ou pour un petit nombre de machines dans un environnement contrôlé.

b) Splashtop

Splashtop est un logiciel de prise en main à distance permettant à un poste client de se connecter à un ordinateur ou serveur cible via réseau local ou Internet. La solution offre une interface graphique complète, sécurisée et multi-plateforme.

- Principe de fonctionnement : l'utilisateur installe le logiciel Splashtop sur la machine distante et sur le poste client. La connexion se fait via l'interface du client, qui communique avec la machine distante pour afficher son bureau et permettre le contrôle à distance.
- Avantages :
    - Interface graphique intuitive, adaptée aux utilisateurs non techniques.
    - Multi-plateforme (Windows, Linux, macOS, iOS, Android).
    - Connexions sécurisées grâce au chiffrement intégré.
    - Gestion simple des accès et des sessions à distance.
- Points à considérer :
    - Nécessite l'installation du logiciel sur le poste client et sur la machine distante.
    - Dépendance à un service tiers, et certaines fonctionnalités peuvent nécessiter une licence payante.
    - Moins flexible pour un environnement entièrement internalisé ou pour une gestion centralisée des accès multiples.

Cette solution est particulièrement adaptée pour un accès distant simple et rapide, avec une interface graphique complète et sécurisée, sans configuration réseau complexe côté client.

c) TeamViewer

TeamViewer est un logiciel de prise en main à distance permettant à un poste client de se connecter à un ordinateur ou serveur cible via Internet ou réseau local. Il fournit une interface graphique complète et sécurisée pour le contrôle à distance.

- Principe de fonctionnement : l'utilisateur installe TeamViewer sur la machine distante et sur le poste client. La connexion s'établit en saisissant un identifiant et un mot de passe fournis par la machine cible, permettant le contrôle complet du bureau distant.
- Avantages :
    - Multi-plateforme : compatible Windows, Linux, macOS, iOS et Android.
    - Interface intuitive et complète, facile à utiliser pour les utilisateurs non techniques.
    - Gestion centralisée possible via des comptes TeamViewer pour plusieurs machines et utilisateurs.
    - Connexions sécurisées grâce au chiffrement intégré.
- Points à considérer :
    - Nécessite l'installation du logiciel sur le poste client et sur la machine distante.
    - Dépendance à un service tiers et certaines fonctionnalités nécessitent une licence payante.
    - La connexion peut dépendre de la qualité de la connexion Internet si utilisée hors réseau local.

Cette solution est adaptée pour un accès distant sécurisé et rapide, avec une interface graphique complète et une gestion des sessions simplifiée, même pour plusieurs machines.

d) Remmina

Remmina est un client open-source destiné principalement aux postes Linux, permettant la prise en main à distance de machines Windows ou Linux via différents protocoles tels que RDP, VNC ou SSH. Il offre une interface graphique pour gérer les connexions à distance.

- Principe de fonctionnement : l'utilisateur installe Remmina sur le poste client. Depuis ce client, il peut se connecter aux machines cibles en utilisant les protocoles supportés, en saisissant les identifiants nécessaires à chaque connexion.
- Avantages :
    - Gratuit et open-source.
    - Multi-protocole : RDP, VNC et SSH.
    - Compatible avec différentes plateformes, mais surtout utilisé sous Linux.
    - Permet de gérer plusieurs connexions dans une seule interface.
- Points à considérer :
    - Nécessite l'installation du client sur le poste utilisateur.
    - Chaque connexion doit être configurée manuellement ; moins centralisé que Guacamole.
    - Fonctionnalités limitées par rapport aux solutions centralisées comme Apache Guacamole ou TeamViewer.

Remmina est particulièrement adapté pour les environnements Linux, permettant un accès multi-protocole aux machines distantes, avec une interface simple et légère pour gérer plusieurs connexions.

e) Apache Guacamole

Apache Guacamole est une solution de prise en main à distance centralisée et accessible via un navigateur web, permettant d'accéder à des machines Windows ou Linux sans installer de client spécifique sur le poste utilisateur. Il supporte plusieurs protocoles, notamment RDP, VNC et SSH.

- Principe de fonctionnement : un serveur Guacamole est installé sur une machine dédiée. Les connexions vers les serveurs cibles (Linux ou Windows) sont gérées par le serveur. Le poste client se connecte simplement via un navigateur web pour accéder à l'interface centralisée et aux machines distantes.
- Avantages :
    - Accès centralisé via navigateur, sans installation de logiciel supplémentaire côté client.
    - Multi-protocole : RDP, VNC, SSH.
    - Gestion centralisée des utilisateurs et des connexions, facilitant l'administration.
    - Permet de gérer plusieurs machines et connexions depuis une seule interface.
- Points à considérer :
    - Nécessite l'installation et la configuration d'un serveur Guacamole et d'une base de données pour gérer les comptes.
    - Configuration initiale plus complexe que pour un client direct comme Remmina.

Cette solution est particulièrement adaptée pour un environnement où plusieurs machines doivent être accessibles à distance de manière centralisée, avec un contrôle fin des utilisateurs et des sessions via une interface web unique.

2- Comparaisons des solutions

Critère	Connexion directe	Splashtop	TeamViewer	Remmina	Apache Guacamole
Type de connexion	Directe via protocole natif	Logiciel propriétaire	Logiciel propriétaire	Client open-source	Serveur centralisé via navigateur
Installation côté client	Nécessaire seulement pour le protocole natif	Oui	Oui	Oui	Non, accès via navigateur
Installation côté serveur / machine distante	Nécessaire pour RDP/SSH activé	Oui	Oui	Dépend du protocole (RDP/VNC/SSH)	Oui (serveur Guacamole)
Multi-protocole	RDP ou SSH	Oui (RDP/VNC/SSH limité selon version)	Oui (RDP/VNC/SSH via extensions)	Oui (RDP, VNC, SSH)	Oui (RDP, VNC, SSH)
Interface graphique	Oui pour RDP, non pour SSH natif	Oui	Oui	Oui	Oui via navigateur
Gestion centralisée des connexions	Non	Limitée	Oui, via compte TeamViewer	Non	Oui, toutes les machines accessibles depuis une interface web unique
Sécurité	Dépend du protocole et des pare-feux	Chiffrement intégré	Chiffrement intégré	Dépend du protocole	Chiffrement intégré, contrôle d'accès centralisé
Facilité d'utilisation	Moyenne, dépend du protocole	Facile	Très facile	Moyenne	Moyenne, configuration initiale plus complexe
Licence / coût	Gratuit	Gratuit ou payant selon version	Gratuit pour usage personnel, payant pour pro	Gratuit et open-source	Gratuit et open-source
Adapté pour environnements multiples / centralisés	Non	Partiellement	Oui	Non	Oui

3- Choix d'une solution

Après analyse des différentes solutions de prise en main à distance, la solution retenue pour ce TP est Apache Guacamole.

Le choix se justifie par plusieurs caractéristiques clés :

- Centralisation des accès : toutes les machines cibles (Linux et Windows) sont accessibles depuis une interface web unique, simplifiant la gestion des connexions et des utilisateurs.
- Multi-protocole : Guacamole supporte RDP, SSH et VNC, permettant de gérer différents types de machines avec une seule solution.
- Facilité d'utilisation côté client : aucun logiciel spécifique n'est nécessaire sur le poste client, qui n'a besoin que d'un navigateur web pour se connecter aux serveurs distants.
- Gestion des utilisateurs et sécurité : les comptes et permissions sont administrables de manière centralisée, offrant un contrôle fin des accès. Les connexions sont sécurisées grâce au chiffrement intégré et aux protocoles standard.
- Gratuité et open-source : Guacamole est libre, ce qui permet de l'utiliser et de le déployer sans coût supplémentaire, tout en restant modulable et personnalisable selon les besoins de l'infrastructure.

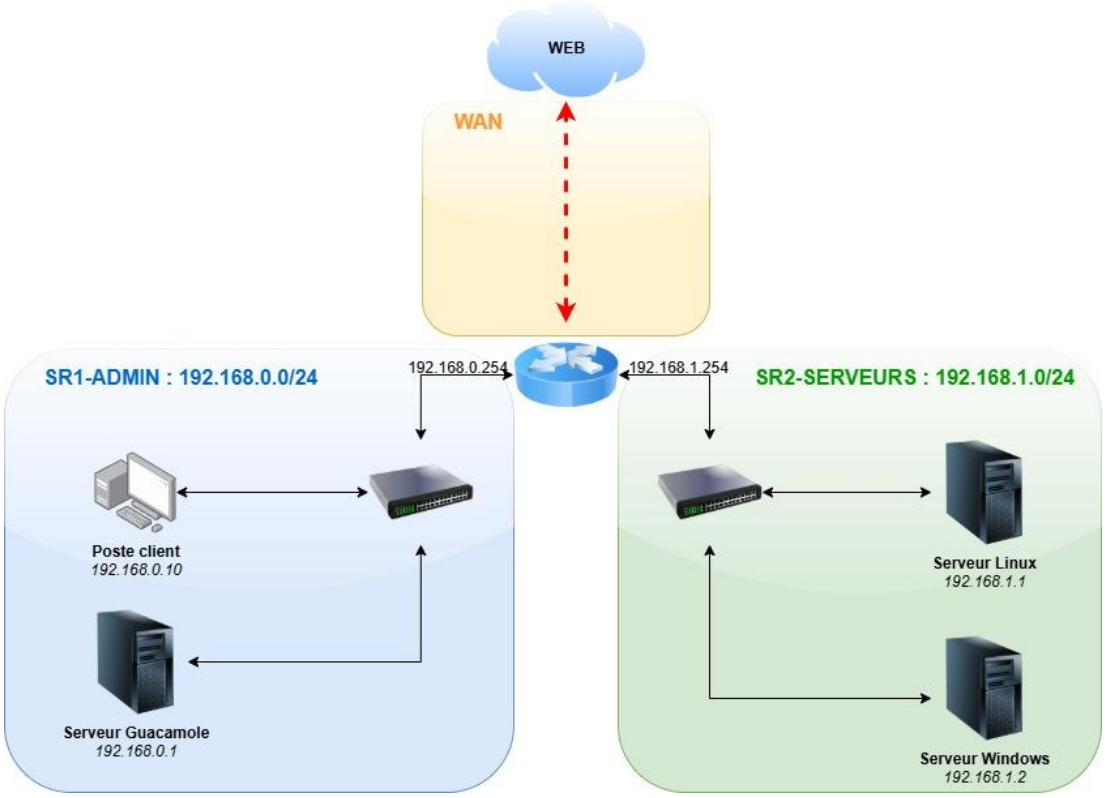
D'autres solutions comme Splashtop ou TeamViewer auraient pu être adaptées pour ce TP, car elles offrent également une interface graphique et des fonctionnalités sécurisées pour la prise en main à distance. Cependant, ces solutions sont payantes pour un usage professionnel ou multi-machines, ce qui les rend moins adaptées dans le cadre d'un TP ou d'un environnement d'apprentissage.

Les solutions comme la connexion directe RDP/SSH ou Remmina restent fonctionnelles, mais elles ne permettent pas une gestion centralisée des accès ni un contrôle simplifié pour plusieurs machines simultanément.

Ainsi, Apache Guacamole a été choisi pour ce TP afin de reproduire un environnement d'entreprise sécurisé, multi-plateforme et centralisé, tout en garantissant une solution gratuite et open-source pour la prise en main à distance.

4- Cartographie du réseau

a) Schéma



b) Plan d'adressage

Réseau	192.168.0.0/24
Masque	255.255.255.0
Serveur Guacamole	192.168.0.1
Poste-Client	192.168.0.10
Passerelle	192.168.0.254
Broadcast	192.168.0.255

Réseau	192.168.1.0/24
Masque	255.255.255.0
Serveur Linux	192.168.1.1
Serveur Windows	192.168.1.2
Passerelle	192.168.1.254
Broadcast	192.168.1.255

c) Tables de routage

Réseau	Adresse	Passerelle	Interface
SR1	192.168.0.0/24	192.168.0.254	192.168.0.254
SR2	192.168.1.0/24	192.168.1.254	192.168.1.254
Défaut	0.0.0.0/0	FAI	WAN

5- Etude de l'impact sur le SI existant

a) Sécurité

La mise en place d'Apache Guacamole dans cette infrastructure de test a plusieurs impacts sur la sécurité du système d'information.

L'un des principaux avantages de Guacamole réside dans la centralisation des accès distants. Plutôt que d'exposer directement les ports SSH et RDP de chaque serveur vers l'extérieur, seul le serveur Guacamole est accessible depuis le réseau client. Cette approche réduit considérablement la surface d'attaque en limitant le nombre de services exposés. Les utilisateurs s'authentifient via une interface web unique, et leurs permissions peuvent être gérées de manière centralisée depuis la base de données de Guacamole. Cette gestion simplifiée permet de contrôler précisément qui accède à quelles ressources et facilite la révocation des accès en cas de besoin.

Les communications entre le poste client et Guacamole peuvent être sécurisées via HTTPS, garantissant le chiffrement des données en transit. Les connexions entre Guacamole et les serveurs cibles utilisent les protocoles natifs SSH et RDP, qui intègrent également leurs propres mécanismes de chiffrement. Cette double couche de protection assure la confidentialité des échanges sur l'ensemble de la chaîne de connexion.

La segmentation réseau entre le réseau client (192.168.0.0/24) et le réseau serveurs (192.168.1.0/24) renforce également la sécurité. Le pare-feu pfSense contrôle les flux entre ces deux réseaux et autorise uniquement le serveur Guacamole à initier des connexions SSH et RDP vers les serveurs cibles. Le poste client Windows 10 ne peut pas accéder directement à ces serveurs, ce qui limite les risques en cas de compromission. Guacamole enregistre par ailleurs toutes les connexions des utilisateurs, offrant une traçabilité complète des accès distants et facilitant les investigations en cas d'incident.

Toutefois, le serveur Guacamole devient un point critique de l'infrastructure, car sa compromission donnerait accès à l'ensemble des serveurs internes. Il est donc essentiel de maintenir le serveur à jour, de sécuriser la base de données contenant les identifiants, et de surveiller régulièrement les logs d'accès. Dans un environnement de production, l'accès à l'interface web devrait être protégé par des mots de passe robustes et, si possible, par une authentification multi-facteurs.

b) Performance

La mise en place d'Apache Guacamole introduit une couche intermédiaire entre le poste client et les serveurs cibles, ce qui peut impacter les performances.

Le serveur Guacamole agit comme un relais qui reçoit les connexions des utilisateurs via le navigateur web, puis établit les sessions SSH ou RDP vers les serveurs distants. Toutes les données échangées transitent par Guacamole, ce qui peut introduire une légère latence supplémentaire par rapport à une connexion directe, notamment lors de l'affichage de contenus graphiques via RDP.

Dans cette infrastructure de test entièrement virtualisée, les performances dépendent des ressources allouées au serveur Guacamole et de la capacité du poste hôte. Le serveur dispose de 2 Go de mémoire et un cœur CPU, ce qui est suffisant pour gérer quelques connexions simultanées dans un environnement de test. En production avec de nombreux utilisateurs, il faudrait augmenter ces ressources pour maintenir des performances acceptables.

Pour cette infrastructure de test, les performances sont largement suffisantes pour les opérations d'administration courantes. En production, il conviendrait de dimensionner le serveur selon le nombre d'utilisateurs simultanés et de surveiller l'utilisation des ressources.

c) Ergonomie

La mise en place d'Apache Guacamole améliore l'ergonomie de l'accès distant par rapport aux solutions traditionnelles.

L'un des principaux avantages de Guacamole réside dans son accessibilité via un simple navigateur web. Les utilisateurs n'ont pas besoin d'installer de logiciel spécifique sur leur poste de travail, ce qui simplifie le déploiement et l'utilisation. Il suffit de se connecter à l'interface web pour accéder à l'ensemble des serveurs distants configurés. Cette approche est particulièrement avantageuse dans les environnements où les postes clients sont verrouillés ou où l'installation de logiciels est restreinte.

L'interface présente une page d'accueil claire affichant toutes les connexions disponibles sous forme de vignettes. L'utilisateur peut se connecter rapidement au serveur souhaité en un simple clic, évitant ainsi de jongler entre plusieurs clients SSH ou RDP. Une fois la session ouverte, l'expérience est proche d'une connexion directe, avec un bureau distant pour RDP ou un terminal pour SSH. Un menu latéral accessible permet d'accéder à des fonctionnalités supplémentaires comme le partage de fichiers ou le presse-papier partagé.

Cependant, l'utilisation d'une interface web peut introduire une légère latence dans l'affichage, notamment pour les connexions RDP avec des contenus graphiques. Certains raccourcis clavier peuvent également entrer en conflit avec ceux du navigateur. Dans le cadre de cette infrastructure de test, Guacamole offre une ergonomie satisfaisante pour les opérations d'administration courantes, avec une interface intuitive adaptée aussi bien aux environnements de test qu'aux déploiements en production.

6- Phasage de l'intervention

La mise en place de l'infrastructure d'accès distant avec Apache Guacamole se déroule en plusieurs phases successives, permettant une progression logique et méthodique du projet.

- **Phase 1 : Préparation de l'environnement de virtualisation :**

Cette première phase consiste à préparer l'environnement de travail sur le poste hôte. VMware Workstation est configuré pour créer les réseaux virtuels nécessaires, correspondant aux deux segments réseau définis dans le plan d'adressage (192.168.0.0/24 et 192.168.1.0/24). Les images ISO de pfSense, Debian 13, Windows Server 2022 et Windows 10 sont récupérées et vérifiées avant le déploiement des machines virtuelles.

- **Phase 2 : Déploiement des machines virtuelles :**

Les cinq machines virtuelles sont créées et configurées avec les ressources allouées définies précédemment. Chaque VM est installée à partir de son image ISO respective. Les systèmes d'exploitation sont installés avec les configurations de base, et les adresses IP statiques sont attribuées selon le plan d'adressage établi. Cette phase inclut également la configuration initiale des comptes administrateurs et la mise à jour des systèmes.

- **Phase 3 : Configuration du pare-feu pfSense :**

Le pare-feu pfSense est configuré pour assurer le routage entre les deux réseaux internes et l'accès Internet. Les interfaces SR1 (192.168.0.254) et SR2 (192.168.1.254) sont paramétrées, ainsi que l'interface WAN pour l'accès Internet. Les règles de pare-feu sont mises en place pour autoriser les flux nécessaires entre les réseaux tout en maintenant un niveau de sécurité adapté. Le serveur DHCP est désactivé car toutes les machines utilisent des adresses IP statiques.

- **Phase 4 : Installation et configuration du serveur Apache Guacamole :**

Cette phase constitue le cœur du projet. Le serveur Debian 13 hébergeant Guacamole est préparé avec l'installation des prérequis nécessaires, notamment le serveur web Apache ou Tomcat, une base de données MariaDB ou MySQL pour stocker les configurations et les utilisateurs, ainsi que les dépendances système requises. Apache Guacamole est ensuite installé et configuré. La base de données est initialisée avec le schéma fourni par Guacamole, et l'accès HTTPS est configuré pour sécuriser les connexions web.

- **Phase 5 : Configuration des serveurs cibles :**

Les serveurs Linux et Windows Server 2022 sont préparés pour accepter les connexions distantes. Sur le serveur Debian 13 cible, le service SSH est activé et configuré pour autoriser les connexions depuis le serveur Guacamole. Sur le serveur Windows Server 2022, le Bureau à distance (RDP) est activé et configuré avec les autorisations nécessaires. Des comptes utilisateurs sont créés sur chaque serveur pour permettre les connexions via Guacamole.

- **Phase 6 : Configuration des connexions dans Guacamole :**

Les connexions vers les serveurs cibles sont créées dans l'interface d'administration de Guacamole. Pour chaque serveur, une connexion est configurée avec les paramètres appropriés : protocole (SSH ou RDP), adresse IP du serveur cible, port de connexion, et identifiants d'accès. Des utilisateurs sont créés dans Guacamole et des permissions leur sont attribuées pour accéder aux connexions configurées.

- **Phase 7 : Tests et validation :**

Cette dernière phase consiste à valider le bon fonctionnement de l'ensemble de l'infrastructure. Depuis le poste client Windows 10, les connexions sont testées via l'interface web de Guacamole pour vérifier l'accès aux serveurs Linux et Windows. Les tests incluent la vérification de l'authentification, de la qualité de la connexion, du transfert de fichiers, et de la stabilité des sessions. Les éventuels dysfonctionnements sont identifiés et corrigés avant la mise en production de l'infrastructure de test.

7- Prévision des tests de validation

Les tests de validation permettent de s'assurer que l'infrastructure fonctionne correctement et que les objectifs définis dans le cahier des charges sont atteints.

Tests de connectivité réseau :

Des tests ping sont effectués depuis chaque machine vers les passerelles et les autres machines des deux réseaux pour confirmer la configuration réseau et le routage via pfSense. L'accès Internet est également vérifié depuis chaque machine pour valider la configuration WAN du pare-feu.

Tests d'accès au serveur Guacamole :

Depuis le poste client Windows 10, l'accès à l'interface web de Guacamole est testé via le navigateur. L'authentification est vérifiée avec les identifiants créés dans la base de données. Après connexion, l'interface doit afficher les connexions disponibles pour l'utilisateur selon les permissions attribuées.

Tests de connexion SSH et RDP :

Une connexion SSH vers le serveur Debian 13 (192.168.1.1) est initiée depuis Guacamole pour vérifier que le terminal s'affiche correctement dans le navigateur. Des commandes de base sont exécutées pour tester la réactivité de la session. De même, une connexion RDP vers le serveur Windows Server 2022 (192.168.1.2) est testée pour s'assurer que le bureau Windows s'affiche et que les interactions fonctionnent correctement. Le partage de fichiers et le presse-papier partagé sont également vérifiés.

Tests de sécurité et permissions :

Différents comptes utilisateurs sont créés avec des niveaux de permissions variés pour vérifier que chaque utilisateur accède uniquement aux connexions autorisées. Des tentatives de connexion directe aux serveurs cibles depuis le poste client sont effectuées pour confirmer que les règles du pare-feu bloquent bien les accès directs. Seul le serveur Guacamole doit pouvoir se connecter aux serveurs du réseau 192.168.1.0/24.

Tests de stabilité :

Des sessions prolongées sont maintenues pour vérifier la stabilité de l'infrastructure. L'utilisation des ressources (CPU, mémoire) est surveillée sur le serveur Guacamole pour identifier d'éventuels problèmes de performance.

8- Déploiement

Le déploiement s'effectue en plusieurs étapes successives. VMware Workstation est d'abord configuré avec les deux réseaux virtuels (192.168.0.0/24 et 192.168.1.0/24). Les cinq machines virtuelles sont ensuite créées et installées avec leurs systèmes d'exploitation respectifs.

Les adresses IP statiques sont configurées selon le plan d'adressage établi. pfSense est paramétré pour assurer le routage entre les deux réseaux et les règles de pare-feu nécessaires sont mises en place.

Sur le serveur Debian 13 (192.168.0.1), Apache Guacamole est installé avec ses dépendances (Tomcat, MariaDB, guacd). La base de données est initialisée et l'application web est déployée. Les serveurs cibles sont préparés avec l'activation de SSH sur le serveur Linux et de RDP sur le serveur Windows.

Les connexions SSH et RDP sont configurées dans l'interface d'administration de Guacamole avec les identifiants appropriés. Des comptes utilisateurs sont créés avec leurs permissions d'accès. Enfin, des tests de validation sont effectués pour vérifier le bon fonctionnement de l'ensemble.

IV/ Mise en place

1- Réalisation

a) Création des VMs

- Installation de PFSense

Rappel :

- 256 Mo de mémoire
- 5Go de disque
- 1 cœur
- 3 interfaces (WAN, SR1 et SR2)

Ne pas oublier d'attribuer manuellement la troisième interface !

Configuration réseau :

- WAN : DHCP
- SR1 : 192.168.0.254/24
- SR2 : 192.168.1.254/24

- Installation de Debian 13 sans interface graphique

Rappel :

- 2 Go de mémoire
- 20 Go de disque
- 1 cœur

Deux VMs : une pour le serveur Guacamole et une pour le serveur cible.

Configuration réseau :

- Serveur Guacamole : 192.168.0.1
  - *Masque : 255.255.255.0*
  - *Passerelle : 192.168.0.254*
- Serveur cible : 192.168.1.1
  - *Masque : 255.255.255.0*
  - *Passerelle : 192.168.1.254*

- Installation de Windows server 2022

Rappel :

- 4 Go de mémoire
- 40 Go de disque
- 2 cœurs

Configuration réseau :

- 192.168.1.2
  - *Masque : 255.255.255.0*
  - *Passerelle : 192.168.1.254*

- Installation de Windows 10

  Rappel :
  - 4 Go de mémoire
  - 40 Go de disque
  - 2 coeurs

  Configuration réseau :

  - 192.168.0.10
    - *Masque : 255.255.255.0*
    - *Passerelle : 192.168.0.254*

b) Configuration du pare-feu

Configuration des interfaces réseau :

Le pare-feu pfSense dispose de trois interfaces réseau configurées :

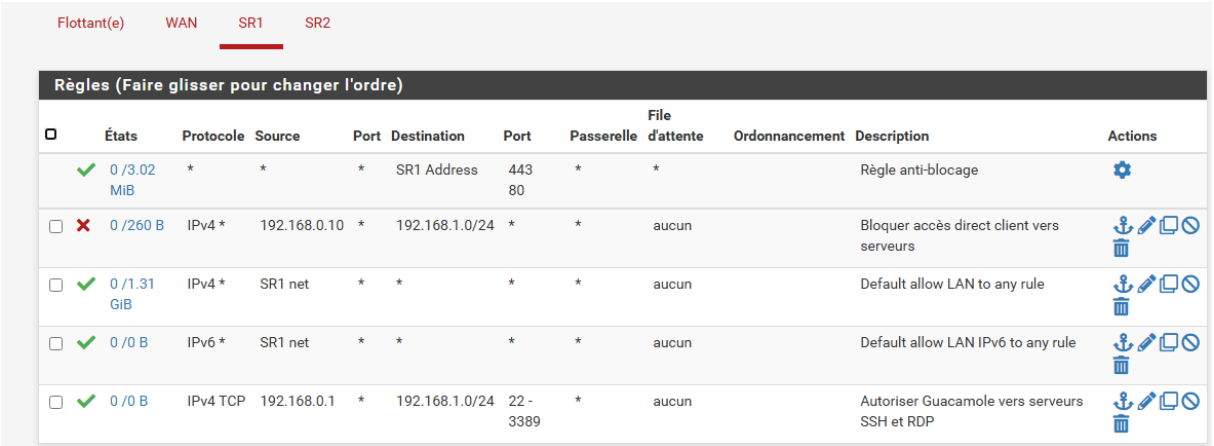
- WAN : interface connectée à Internet, obtenant une adresse IP via DHCP
- SR1 : interface du réseau client/Guacamole avec l'adresse 192.168.0.254/24
- SR2 : interface du réseau serveurs avec l'adresse 192.168.1.254/24

Configuration des règles de pare-feu :

- Règles appliquées sur l'interface SR1 (192.168.0.0/24)

Trois règles ont été créées pour contrôler le trafic du réseau client :

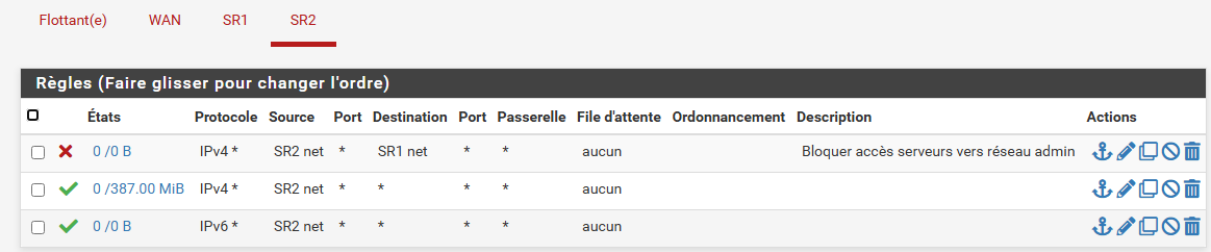
1) Blocage des accès directs du poste client vers les serveurs : cette règle interdit au poste client Windows 10 (192.168.0.10) d'accéder directement au réseau serveurs (192.168.1.0/24). Cela force l'utilisateur à passer obligatoirement par Guacamole pour accéder aux serveurs cibles.
2) Autorisation de Guacamole vers les serveurs : cette règle autorise uniquement le serveur Guacamole (192.168.0.1) à initier des connexions TCP vers le réseau serveurs (192.168.1.0/24) sur les ports 22 (SSH) et 3389 (RDP). Cette règle permet à Guacamole d'établir les sessions distantes vers les serveurs Linux et Windows.
3) Autorisation de l'accès Internet : cette règle autorise l'ensemble du réseau SR1 à accéder à Internet, permettant ainsi les mises à jour système et l'accès aux ressources externes nécessaires.



- Règles appliquées sur l'interface SR2 (192.168.1.0/24)

Deux règles ont été créées pour le réseau serveurs :

1) Blocage des accès des serveurs vers le réseau admin : cette règle de sécurité empêche les serveurs du réseau SR2 d'initier des connexions vers le réseau SR1. Cela limite les risques en cas de compromission d'un serveur et respecte le principe de segmentation réseau.
2) Autorisation de l'accès Internet : cette règle autorise l'ensemble du réseau SR2 à accéder à Internet, permettant les mises à jour et l'installation de paquets nécessaires.



Principe de fonctionnement :

L'architecture mise en place repose sur une segmentation stricte des flux réseau. Le poste client ne peut pas accéder directement aux serveurs du réseau SR2, ce qui réduit la surface d'attaque. Seul le serveur Guacamole, agissant comme bastion, est autorisé à établir des connexions SSH et RDP vers les serveurs cibles. Les serveurs, quant à eux, ne peuvent pas remonter vers le réseau client, garantissant ainsi une isolation unidirectionnelle des flux. Cette configuration respecte les bonnes pratiques de sécurité en limitant les communications au strict nécessaire tout en permettant l'accès Internet pour les opérations de maintenance.

c) Installation d'Apache Guacamole

Dans le cadre de ce projet, plusieurs tentatives d'installation manuelle d'Apache Guacamole ont été réalisées sur différentes versions de Debian avec compilation depuis les sources. Ces installations ont systématiquement rencontré des problèmes de compatibilité entre les différentes versions des composants :

- Debian 13 avec FreeRDP3 et Tomcat10 : incompatibilité entre Guacamole 1.6.0 (utilisant `javax.servlet.*`) et Tomcat 10 (nécessitant `jakarta.servlet.*` depuis Jakarta EE)
- Debian 12 avec FreeRDP2 et Tomcat 9 : problèmes de dépendances et de versions de bibliothèques
- Compilation manuelle : nécessite l'installation et la configuration de nombreuses dépendances (plus de 20 paquets), avec des risques d'erreurs et de conflits de versions

Face à ces difficultés récurrentes, la solution Docker a été retenue pour les raisons suivantes :

Avantages de Docker pour Guacamole :

- Compatibilité garantie : les images Docker officielles Apache Guacamole contiennent des versions testées et compatibles entre elles (Tomcat, FreeRDP, bibliothèques)
- Installation rapide : déploiement en quelques minutes au lieu de plusieurs heures
- Simplicité de maintenance : mises à jour facilitées par le remplacement des conteneurs
- Isolation : les conteneurs n'interfèrent pas avec le système hôte

- Reproductibilité : la même configuration fonctionne sur n'importe quelle machine équipée de Docker
- Documentation officielle : Apache Guacamole fournit une documentation complète pour le déploiement Docker

Cette approche containerisée est documentée et supportée officiellement par Apache Guacamole comme méthode de déploiement simplifiée, particulièrement adaptée aux environnements de test et aux déploiements rapides. *Source : https://guacamole.apache.org/doc/gug/guacamole-docker.html*

➢ Installation et configuration d'Apache Guacamole

2- Rapport de tests

L'ensemble des tests de validation a été réalisé pour vérifier le bon fonctionnement de l'infrastructure d'accès distant et s'assurer que les objectifs du cahier des charges sont atteints. Ces tests couvrent la connectivité réseau, la configuration du pare-feu, l'accès à l'interface Guacamole et les connexions aux serveurs cibles.

Tests de connectivité réseau et sécurité :

Les premiers tests ont porté sur la vérification de la connectivité réseau entre le serveur Guacamole et les serveurs cibles du réseau 192.168.1.0/24. Depuis les conteneurs Docker, des tests de connectivité ont été effectués vers le port SSH (22) du serveur Linux et le port RDP (3389) du serveur Windows. Ces tests ont confirmé que le routage via pfSense fonctionne correctement et que les conteneurs peuvent communiquer avec les serveurs du réseau cible.

Pour valider l'efficacité des règles de pare-feu, des tentatives de connexion directe ont été effectuées depuis le poste client Windows 10 vers les serveurs cibles. Ces tentatives ont été bloquées conformément à l'architecture de sécurité mise en place, confirmant que seul le serveur Guacamole est autorisé à initier des connexions vers les serveurs du réseau 192.168.1.0/24. Le poste client ne peut accéder aux ressources internes qu'en passant par l'interface web Guacamole, garantissant ainsi la centralisation et le contrôle des accès distants.

Tests d'accès à l'interface web Guacamole :

L'accès à l'interface web de Guacamole a été testé depuis le poste client via l'URL http://192.168.0.1:8080/guacamole. La page de connexion s'est affichée correctement, confirmant que le serveur web Tomcat fonctionne et que l'application Guacamole est bien déployée. L'authentification avec les identifiants par défaut a fonctionné immédiatement, et la modification du mot de passe administrateur pour des raisons de sécurité s'est effectuée sans problème. La reconnexion avec le nouveau mot de passe a validé que le système d'authentification MySQL enregistre correctement les modifications dans la base de données.

Tests de connexion SSH vers le serveur Linux :

La configuration d'une connexion SSH vers le serveur Debian Linux a été réalisée dans l'interface Guacamole avec les paramètres nécessaires : adresse IP 192.168.1.1, port 22, et les identifiants de connexion. La première tentative de connexion a échoué avec le message "Connexion échouée". L'analyse des logs Guacamole a révélé que les connexions se fermaient immédiatement après environ

deux secondes, indiquant un problème d'authentification plutôt qu'un problème de connectivité réseau.

Le diagnostic a permis d'identifier que la connexion SSH en tant que root était désactivée par défaut dans le fichier de configuration du serveur Linux. Après modification du fichier /etc/ssh/sshd_config pour autoriser la connexion root avec le paramètre PermitRootLogin yes et redémarrage du service SSH, la connexion a fonctionné immédiatement. Le terminal SSH du serveur Linux s'est affiché correctement dans le navigateur, permettant l'exécution de commandes sans latence notable. Les tests de fonctionnalités avancées, incluant l'exécution de commandes système, la navigation dans l'arborescence de fichiers, l'édition de fichiers avec nano et le copier-coller entre le navigateur et le terminal, ont tous été validés avec succès. Cette méthode de modification du fichier sshd.config pour autoriser une connexion en SSH directement avec le superutilisateur root est déconseillé dans un environnement de production. Il convient de créer un utilisateur dédié à la connexion en SSH puis, éventuellement, de se connecter en root.

Tests de connexion RDP vers le serveur Windows :

La configuration d'une connexion RDP vers le serveur Windows Server 2022 a été créée dans Guacamole avec les paramètres appropriés. Les premières tentatives de connexion ont rencontré plusieurs difficultés techniques qui ont nécessité un diagnostic approfondi. L'analyse des logs du conteneur guacd a révélé différents messages d'erreur au fil des tentatives de résolution.

La première erreur identifiée était "Certificate validation failed", indiquant que le certificat SSL auto-signé du serveur Windows était rejeté par Guacamole. L'activation de l'option "Ignore server certificate" dans les paramètres de sécurité de la connexion a permis de contourner cette vérification, mais la connexion échouait toujours avec un nouveau message d'erreur : "Server refused connection (wrong security type?)". Ce message indiquait que le mode de sécurité configuré ne correspondait pas à celui attendu par le serveur Windows.

Après plusieurs tests avec différents modes de sécurité (RDP, NLA, Any), la solution finale a consisté à configurer le mode de sécurité sur TLS tout en maintenant l'option "Ignore server certificate" activée. Cette configuration a immédiatement permis d'établir la connexion RDP. Le bureau Windows Server 2022 s'est alors affiché correctement dans le navigateur, avec des interactions souris et clavier fonctionnelles et une qualité d'affichage satisfaisante pour un environnement de test virtualisé.

Pour confirmer que le problème ne venait pas du serveur Windows lui-même mais bien de la configuration Guacamole, un test de validation a été effectué en désactivant temporairement la règle de pare-feu bloquant les accès directs du poste client. La connexion RDP directe depuis le poste client Windows 10 vers le serveur Windows a fonctionné immédiatement, validant ainsi que le service RDP était correctement configuré et que le problème résidait uniquement dans les paramètres de sécurité de Guacamole. Après réactivation de la règle de pare-feu, les accès directs ont été à nouveau bloqués comme prévu.

Les tests de fonctionnalités RDP avancées ont inclus la navigation dans l'explorateur Windows, l'ouverture d'applications telles que le Gestionnaire de serveur et PowerShell, le redimensionnement de la fenêtre RDP et le copier-coller de texte entre le navigateur et le bureau Windows. La session est restée stable pendant toute la durée des tests, sans déconnexion intempestive. Quelques latences

mineures ont été observées lors de l'affichage d'éléments graphiques complexes, ce qui reste acceptable pour un environnement de test entièrement virtualisé.

3- Rapport de déploiement

Le déploiement d'Apache Guacamole a été réalisé sur une machine virtuelle Debian via Docker. Ce choix a permis d'éviter les problèmes de compatibilité rencontrés lors des installations manuelles et d'assurer une mise en place rapide et fiable.

L'architecture repose sur plusieurs conteneurs (Guacamole, guacd et base de données), orchestrés avec Docker Compose. Une fois les conteneurs lancés, l'interface web est immédiatement accessible et fonctionnelle.

Les tests post-déploiement confirment le bon fonctionnement de l'authentification et des connexions SSH et RDP vers les serveurs cibles. Le déploiement est conforme au cahier des charges et opérationnel.

V/ Bilan

1- Conclusion

Ce compte rendu présente la mise en place d'un serveur Apache Guacamole dans un environnement de test entièrement virtualisé. Il ne s'appuie ni sur un TP réalisé en cours, ni sur une situation professionnelle vécue en entreprise, mais sur une démarche personnelle d'expérimentation.

L'objectif était de tester l'installation et la mise en place de la solution Guacamole en tant que bastion afin de pouvoir la reproduire et l'intégrer dans mon projet d'infrastructure de réseau de l'épreuve E6. Cette approche m'a permis de mettre en pratique des notions de virtualisation, de sécurité réseau, de segmentation, ainsi que de déploiement de services via Docker.

Ce travail constitue ainsi une base technique solide et réutilisable pour de futurs projets, tout en se rapprochant des problématiques et des solutions rencontrées en environnement professionnel.

2- Auto-évaluation

Cette expérience m'a permis de consolider mes compétences en virtualisation, en administration de serveurs Linux et Windows, ainsi qu'en configuration de services réseau sécurisés. J'ai appris à :

- Installer et configurer Apache Guacamole via Docker.
- Gérer la communication entre réseaux segmentés à l'aide d'un pare-feu pfSense.
- Tester et valider des connexions SSH et RDP dans un environnement centralisé.
- Identifier et résoudre des problèmes de compatibilité et de sécurité liés aux certificats et aux protocoles.

Je considère avoir atteint les objectifs fixés pour ce projet d'expérimentation. Le travail m'a également aidé à développer mon autonomie et ma capacité à documenter et organiser une procédure technique de manière claire.