

RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
https://www.ifri-uac.bj Courriel : contact@ifri.uac.bj



MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Sécurité Informatique

Présenté par :
Othniel Tayo **ADISSO**

Exploitation du port forwarding comme application spécifique des techniques de tunneling pour optimiser les communications réseau

Sous la supervision :
M. Hervé **AKAKPO**

Membres du jury :

ASSOGBA Emery	Docteur	IFRI	Président
HOUNZANME Vladimir	Ingénieur	IFRI	Examinateur
AKAKPO Hervé	Ingénieur	IFRI	Rapporteur

Année Académique : **2023-2024**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des tableaux viii

Liste des acronymes ix

Glossaire 1

Introduction 2

1 Revue de littérature 4

2 Automatisation du port forwarding 14

3 Test et fonctionnement du script de redirection de port 23

Conclusion 33

Perspectives 34

Bibliographie 35

Webographie 36

Annexe 37

Table des matières 39

Dédicace

Je dédie ce mémoire à :

- mon père Josué ADISSO ;
- ma mère Juanita KOUZOUNHOUE ;
- ma soeur et à mes frères ;
- tous mes amis et camarades de classe.

Remerciements

Mes remerciements vont droit à toutes les personnes et institutions qui m'ont soutenu et encouragé dans la rédaction de ce mémoire, notamment :

- Le professeur Eugène C. EZIN, Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI) ;
- M. Hervé AKAKPO, enseignant à l'Institut de Formation et de Recherche en Informatique (IFRI) ;
- M. Jospy K. GOUDALO, pour ses conseils et son assistance ;
- M. Yessoufou LADJOHOUNLOU, pour ses divers apports ;
- Tout le corps professoral de l'IFRI, pour le savoir qu'ils nous ont transmis ;
- À toutes les personnes, de près ou de loin, qui m'ont permis de réaliser ce projet.

Résumé

Ce mémoire explore l'exploitation du port forwarding comme application spécifique des techniques de tunneling pour optimiser les communications réseau. L'étude examine les principes du tunneling et son rôle dans l'amélioration de l'accès sécurisé aux services réseau, en particulier par la redirection de port. Un script automatisé en BASH est proposé pour simplifier la configuration de ces tunnels, tout en garantissant une gestion sécurisée et optimisée des connexions réseau. Les avantages du port forwarding, notamment pour les petites entreprises, sont mis en évidence, offrant une solution flexible pour optimiser les infrastructures réseau existantes. Ce travail ouvre des perspectives pour de futures recherches visant à perfectionner les techniques de tunneling dans un environnement numérique en constante évolution.

Mots clés : Redirection de port.

Abstract

This thesis explores the exploitation of port forwarding as a specific application of tunneling techniques to optimize network communications. The study examines the principles of tunneling and its role in improving secure access to network services, in particular through port forwarding. An automated BASH script is proposed to simplify the configuration of these tunnels, while ensuring secure and optimized management of network connections. The advantages of port forwarding, especially for small businesses, are highlighted, offering a flexible solution to optimize existing network infrastructures. This work opens up avenues for future research aimed at improving tunneling techniques in a constantly evolving digital environment.

Key words: Port forwarding.

Table des figures

1.1 Présentation d'une architecture basée sur la redirection de port[11] 8
1.2 Présentation d'une architecture basée sur la redirection de port local [7] 9
1.3 Présentation d'une architecture basée sur la redirection de port à distance [7] 9
1.4 Présentation d'une architecture basée sur la redirection de port dynamique [7] 10

2.1 Configuration du fichier sshd_config pour la désactivation de l'accès shell 16
2.2 Configuration du fichier sshd_config pour l'activation de l'authentification par clé SSH uniquement 17
2.4 Choix de la redirection de port 18
2.3 Création de paires clés SSH 18
2.5 Illustration de la redirction locale avec le protocole SSH [9] 19
2.6 Script de la redirection de port local 19
2.7 Illustration de la redirection de port dynamique avec le protocole SSH [9] 20
2.8 Script de redirection de port dynamique 21
2.9 Enter Caption 21
2.10 Illustration de la redirection de port distant [9] 22

3.1 Configuration réseau de la machine cliente (client_VPN) 24
3.2 Configuration réseau de la machine intermédiaire (Kali) 24
3.3 Configuration réseau de la machine virtuelle à atteindre (server_VPN) 25
3.4 Test de connexion au serveur FTP depuis la machine intermédiaire 25
3.5 Lancement du script Automator_port_forwarding.sh 26
3.6 Choix de la redirection de port local 26
3.7 Lancement de la redirection de port local 26
3.8 Connexion de la machine cliente au serveur FTP 27
3.9 Analyse réseau à travers l'outil Wireshark 27
3.10 Lancement du conteneur Wordpress 27
3.11 Installation Wordpress 28
3.12 Page de connexion 28
3.13 Execution du script Automator_port_fowrading dans le cas de la redirection dynamique 29
3.14 Paramétrage pour l'utilisation du proxy SOCKS au niveau du navigateur 29
3.15 Accès à la page de connexion depuis la machine cliente (redirection de port dynamique) 30
3.16 Accès à l'application Wordpress depuis la machine cliente (redirection de port dynamique) 30
3.17 Configuration Wordpress 31
3.18 Capture réseau avec Wireshark 31
3.19 Première partie du script (configuration des clés SSH et le choix de la redirection) . . . 37

3.20 Deuxième partie du script (choix de la redirection de port) 37
3.21 Troisième partie du script (configuration des paramètres de la redirection de port local) 38
3.22 Quatrième partie du script (configuration des paramètres de la redirection de port dynamique) 38
3.23 Cinquième partie du script (gestion des entrées utilisateur) 38

Liste des tableaux

1.1 Comparaison des types de port forwarding 10

Liste des acronymes

BASH :

Bourne Again Shell 16

IPSec :

Internet Protocol Security 4

NAT :

Network Address Translation 7

PAT :

Port Address Translation 7

SSH :

Secure Shell 4, 6, 16, 17

SSL :

Secure Socket Layer 5, 6

TLS :

Transport Layer Security 5, 6

Glossaire

Proxy :

Un proxy est un serveur intermédiaire qui agit comme un relais entre un client et un serveur final sur Internet. 20

SOCKS :

Le protocole SOCKS (Socket Secure) est un protocole réseau qui facilite le transfert de données entre un client et un serveur via un serveur proxy. 9, 20

Introduction Générale

L'évolution des technologies de l'information et de la communication a profondément redéfini les modalités de transmission des données dans les réseaux modernes. Dans ce contexte, le tunneling s'impose comme une technique essentielle, regroupant diverses méthodes, dont le port forwarding, pour encapsuler et transmettre des données de manière sécurisée et efficace. Le port forwarding, en particulier, permet de rediriger des connexions réseau vers des destinations spécifiques, tout en assurant la confidentialité et l'intégrité des échanges.
Cette méthode joue un rôle crucial dans de nombreux scénarios, tels que l'accès sécurisé à des services internes, la gestion de connexions distantes ou encore l'interconnexion de ressources au sein d'infrastructures distribuées. Ce mémoire vise à explorer les mécanismes sous-jacents du port forwarding, tout en le replaçant dans le cadre général du tunneling, afin de démontrer sa pertinence et son efficacité dans l'optimisation des réseaux modernes.

1. Contexte

Dans un monde de plus en plus interconnecté, les organisations doivent faire face à des défis croissants liés à la sécurité, à l'accessibilité et à l'optimisation des flux de données. Les méthodes traditionnelles de transmission réseau montrent leurs limites face à des exigences accrues en termes de protection contre les cyberattaques, d'efficacité dans la gestion des ressources et de flexibilité pour les environnements hybrides. Le tunneling, et plus particulièrement le port forwarding, répond à ces enjeux en offrant une solution adaptée pour contrôler et sécuriser le trafic réseau tout en permettant un accès optimisé aux services.

2. Problématique

Dans un environnement réseau en constante évolution, où les besoins en sécurité et en efficacité sont cruciaux, le port forwarding émerge comme une solution stratégique. Comment exploiter le port forwarding, en tant qu'application spécifique des techniques de tunneling, pour optimiser la sécurité, la performance et l'accessibilité des infrastructures réseau dans divers environnements ?

3. Objectifs

L'objectif principal de cette étude est de proposer des solutions basées sur la redirection de port (port forwarding) pour enrichir et optimiser les usages des infrastructures réseau, en garantissant des communications sécurisées et accessibles. Quant aux objectifs spécifiques, nous avons :

- Comprendre les fondements du tunneling et la place du port forwarding dans ce contexte.
- Analyser les avantages et les limitations du port forwarding dans différents scénarios réseau.
- Développer un outil automatisé pour simplifier et sécuriser la mise en œuvre du port forwarding.
- Évaluer l'efficacité de cette approche à travers des cas pratiques et des tests de performance.

4. Structuration du document

Le document s'articule autour de trois (03) chapitres à savoir :

1. **Revue de littérature** :
    - Présentation générale du tunneling et du port forwarding.
    - Classification des types de port forwarding : local, distant et dynamique.
    - Analyse des avantages, des limites et des applications du port forwarding.
2. **Conception d'un script pour l'utilisation du port forwarding** :
    - Outils nécessaires pour l'implémentation.
    - Description des fonctionnalités du script : automatisation, sécurité et flexibilité.
    - Étapes de développement et configurations réseau requises.
3. **Tests et évaluation du script** :
    - Cas d'utilisation pratiques : accès distant sécurisé, gestion des services internes, interconnexion réseau.
    - Analyse des résultats obtenus : gains en performance, robustesse de la sécurité et adaptabilité du script.

Chapitre 1

Revue de littérature

Introduction

Le tunneling, technique permettant d'encapsuler et de sécuriser les données, est devenu essentiel pour répondre aux défis croissants de la communication en réseau. Ce chapitre explore les bases théoriques du tunneling et met en lumière le port forwarding, une méthode clé pour optimiser et sécuriser l'accès aux ressources réseau.

1.1 Introduction au tunneling

1.1.1 Définition et principes fondamentaux

Le tunneling est une technique de communication réseau qui consiste à encapsuler des données dans un autre protocole afin de les transmettre de manière sécurisée ou optimisée à travers un réseau public ou privé. Ce procédé permet de masquer le contenu des données ainsi que les adresses source et destination, offrant ainsi une confidentialité accrue et une protection contre les interceptions ou modifications malveillantes. Cette méthode repose sur l'utilisation de "tunnels", qui sont des canaux virtuels à travers lesquels transitent les données encapsulées. Les informations sont chiffrées avant d'être insérées dans ces tunnels, ce qui garantit leur sécurité tout au long du transfert. Une fois arrivées à destination, les données sont décapsulées et déchiffrées pour être utilisées normalement. [8]
Le tunneling est particulièrement prisé dans des situations où la confidentialité et l'intégrité des données sont essentielles. Il trouve des applications variées, notamment dans la création de connexions sécurisées via des VPN (Virtual Private Networks), l'accès distant à des ressources internes, ou encore la mise en place de solutions de télétravail sécurisées. Les principaux protocoles utilisés pour mettre en œuvre le tunneling sont :

- SSH : Protocole polyvalent permettant d'établir des connexions sécurisées, souvent utilisé pour des tâches comme le port forwarding ou l'administration à distance.
- IPSec : Une suite de protocoles destinée à sécuriser les communications au niveau de la couche réseau, souvent utilisée dans les VPN.

- SSL/TLS : Employé pour sécuriser les connexions sur Internet, notamment dans les transactions bancaires et le commerce électronique.

Ces outils permettent au tunneling de répondre efficacement aux besoins de sécurité et de flexibilité des infrastructures réseau modernes.

1.1.2 Importance du tunneling dans les réseaux modernes

Le tunneling (ou création de tunnels) dans les réseaux modernes est devenu une technologie incontournable, surtout face à l'augmentation des menaces de cybersécurité et à la nécessité de connecter des infrastructures distribuées et éloignées. Il consiste à encapsuler des paquets de données à l'intérieur d'autres paquets afin de garantir la confidentialité, l'intégrité et l'authenticité des informations qui transitent sur des réseaux potentiellement vulnérables ou non sécurisés. Cette méthode est cruciale dans le contexte actuel où les réseaux sont souvent confrontés à des attaques, des fuites de données ou des intrusions malveillantes. Le tunneling remplit plusieurs rôles essentiels :

- Sécuriser les connexions sur des réseaux non fiables : Dans de nombreux cas, les données doivent transiter à travers des réseaux publics ou non sécurisés, comme les hotspots Wi-Fi publics dans les cafés, aéroports, hôtels ou autres lieux publics. Ces réseaux sont souvent vulnérables aux attaques telles que les attaques Man-in-the-Middle (MITM), où un attaquant intercepte et modifie les communications. Le tunneling permet de créer un canal sécurisé entre l'utilisateur et le serveur, empêchant l'écoute clandestine et la manipulation des données en transit. Il garantit que même si un attaquant parvient à accéder au réseau, il ne pourra pas déchiffrer ou altérer les informations échangées.

- Accès distant sécurisé aux ressources internes des entreprises : Les entreprises modernes, souvent réparties sur plusieurs sites géographiques, permettent à leurs employés d'accéder aux ressources internes via des connexions à distance, comme les réseaux privés virtuels (VPN). Le tunneling est l'un des principaux mécanismes utilisés pour établir des connexions sécurisées entre les travailleurs à distance et les réseaux d'entreprise. Grâce au VPN, un utilisateur distant peut accéder aux applications, bases de données et autres ressources de l'entreprise comme s'il se trouvait physiquement dans les locaux, tout en s'assurant que la connexion reste confidentielle et protégée contre toute tentative d'intrusion.

- Optimisation des performances réseau et réduction des congestions : Le tunneling ne se limite pas à la sécurité des données, il peut aussi jouer un rôle dans l'optimisation des performances réseau. En encapsulant les données dans des paquets supplémentaires, le tunneling permet de réduire les congestions du réseau en organisant plus efficacement le trafic. Cela peut être particulièrement utile dans des situations où le réseau subit des charges élevées ou dans des environnements où les données doivent transiter à travers des infrastructures peu fiables. Le tunneling permet aussi d'adopter des stratégies comme le multiplexage, où plusieurs flux de données sont envoyés via un même tunnel, maximisant ainsi l'utilisation de la bande passante disponible.

- Préserver l'intégrité et la confidentialité des données : En plus de garantir la confidentialité (en chiffrant les données), le tunneling veille aussi à l'intégrité des informations en transit. Cela signifie que toute tentative de modification des données est détectée. La combinaison du

chiffrement et des mécanismes de vérification de l'intégrité (tels que les sommes de contrôle et les signatures numériques) assure que les données n'ont pas été corrompues ou altérées durant leur transmission. Ceci est particulièrement important dans les applications sensibles, telles que les transactions financières en ligne ou les échanges d'informations confidentielles dans le cadre professionnel.

Dans un monde hyperconnecté où les menaces évoluent constamment et où les données doivent circuler de manière fluide mais sécurisée à travers des réseaux complexes, le tunneling est devenu un outil indispensable pour garantir la **sécurité**, l'**accessibilité**, et la **performance des échanges de données**. Que ce soit dans un contexte professionnel ou personnel, il permet de répondre à des défis variés tout en offrant des garanties de confidentialité et de fiabilité.

1.1.3 1.1.3 Applications courantes du tunneling (VPN, SSH, etc.)

Le tunneling est au cœur de nombreuses applications réseau utilisées quotidiennement pour sécuriser les échanges de données ou optimiser les communications. Parmi les applications les plus courantes, on trouve :

1. Les réseaux privés virtuels (VPN)
   Les VPN (Virtual Private Networks) exploitent le tunneling pour créer une connexion sécurisée entre un utilisateur et un réseau distant. En encapsulant et en chiffrant les données dans un tunnel virtuel, les VPN garantissent la confidentialité et l'intégrité des informations échangées, même sur des réseaux publics comme les points d'accès Wi-Fi. [8] Les principales utilisations des VPN incluent :

   - La protection des données sensibles lors de connexions à distance.
   - Le contournement des restrictions géographiques pour accéder à du contenu bloqué.
   - La sécurisation des communications internes des entreprises.

2. Le protocole SSH
   SSH est une autre application courante du tunneling. Il permet de sécuriser les connexions entre deux machines en encapsulant les données dans un canal chiffré. [6] SSH est particulièrement utilisé pour des tâches comme :

   - L'administration à distance des serveurs et des systèmes.
   - Le port forwarding (local, distant ou dynamique) pour rediriger le trafic réseau de manière sécurisée.
   - La création de tunnels sécurisés pour des services qui ne supportent pas nativement le chiffrement.

3. SSL/TLS et les applications Web sécurisées
   SSL (Secure Sockets Layer) et TLS (Transport Layer Security) sont des protocoles de tunneling utilisés principalement pour sécuriser les communications sur Internet. Ces protocoles établissent un tunnel sécurisé entre un navigateur web et un serveur pour protéger des transactions sensibles, comme :

   - Les paiements en ligne et les transactions bancaires.
   - Les échanges d'informations personnelles sur des plateformes sécurisées.

- Les communications entre des API et des services cloud.

4. L'utilisation dans les infrastructures distribuées et les environnements cloud
   Dans les infrastructures modernes, telles que les environnements multi-clouds et les systèmes distribués, le tunneling est utilisé pour :

   - Assurer une communication sécurisée entre des services hébergés dans différents environnements.
   - Protéger les connexions entre des points de terminaison dans des réseaux complexes.
   - Faciliter l'intégration de solutions de télétravail avec des infrastructures centralisées.

Ces différentes applications démontrent la polyvalence du tunneling dans les environnements réseau actuels, qu'il s'agisse de protéger des données sensibles, d'optimiser l'accessibilité ou de répondre à des besoins spécifiques de connectivité.

1.2 Présentation du port forwarding (redirection de port

1.2.1 Définition et concepts de base

Le port forwarding, ou redirection de port, est une technique utilisée pour rediriger le trafic réseau reçu sur un port spécifique d'une machine ou d'un équipement réseau vers un autre port ou une autre machine sur le réseau. Cette méthode permet d'accéder à des services ou des ressources hébergées derrière des routeurs, des pare-feu ou d'autres dispositifs intermédiaires, tout en garantissant une communication fluide et sécurisée. Ce concept est similaire au PAT (Port Address Translation), une extension de NAT (Network Address Translation), qui permet de mapper plusieurs appareils d'un réseau local sur une seule adresse IP publique en attribuant des numéros de port distincts.[4]

Le port forwarding fonctionne en créant une sorte de "pont" entre un réseau externe (tel qu'Internet) et un réseau interne, permettant ainsi à des dispositifs ou services internes de recevoir des connexions provenant de l'extérieur. Les principaux objectifs de cette méthode incluent :

- Permettre l'accès à des services internes tels que des serveurs web, FTP ou SSH depuis l'extérieur.
- Simplifier la gestion des accès dans les environnements réseau complexes.
- Renforcer la sécurité en limitant les connexions externes uniquement à des ports spécifiques.

Le port forwarding repose sur des règles prédéfinies qui dictent comment rediriger le trafic. Lorsqu'un paquet arrive à un routeur ou un serveur configuré pour le port forwarding :

1. Le dispositif identifie le port de destination spécifié dans le paquet.
2. Il applique la règle de redirection associée à ce port, modifiant au besoin l'adresse IP de destination et le numéro de port.
3. Le paquet est ensuite acheminé vers sa destination finale sur le réseau interne.

Cette méthode est souvent utilisée en conjonction avec des protocoles de tunneling comme SSH ou des technologies de NAT (Network Address Translation) pour optimiser et sécuriser les connexions.

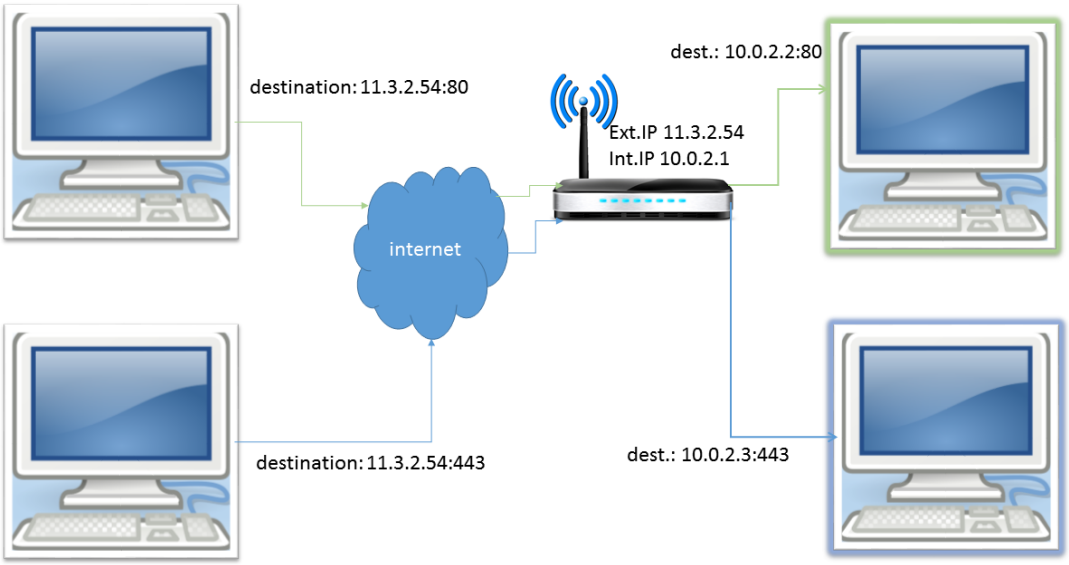


FIGURE 1.1 : Présentation d'une architecture basée sur la redirection de port[11]

1.2.2 Les types de port forwarding

Le port forwarding peut être classé en trois principaux types, chacun ayant des caractéristiques et des usages spécifiques : local, distant et dynamique. Ces variantes permettent de répondre à différents besoins en termes de redirection de trafic réseau.

- **Redirection de port local (Local port forwarding)** : Le port forwarding local est utilisé pour permettre à un utilisateur local d'accéder à des services distants qui ne sont normalement pas accessibles directement depuis son réseau. Cette méthode consiste à rediriger un port de la machine locale vers un port spécifique sur un hôte distant via un intermédiaire, souvent à l'aide d'un tunnel sécurisé comme SSH.[3]
  Fonctionnement
  - L'utilisateur configure un tunnel sur sa machine locale.
  - Le trafic envoyé vers le port spécifié localement est redirigé via le tunnel vers l'hôte et le port cibles.
  - L'utilisateur interagit alors avec le service distant comme s'il était accessible localement.

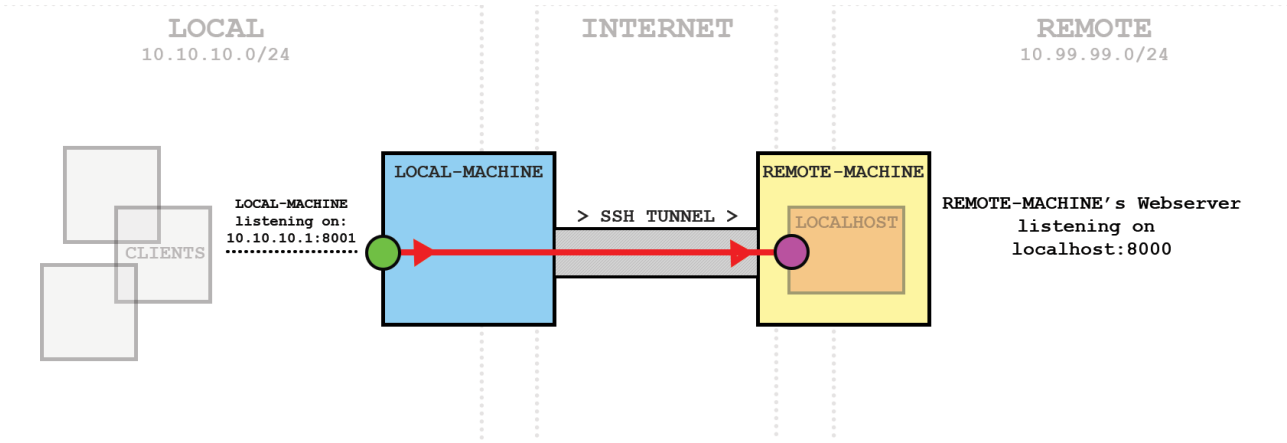


FIGURE 1.2 : Présentation d'une architecture basée sur la redirection de port local [7]

- **Redirection de port à distance (Remote port forwarding)** : Le port forwarding distant, quant à lui, permet à des utilisateurs ou services externes d'accéder à une ressource locale en passant par un intermédiaire. Ce type de redirection est particulièrement utile pour rendre accessibles des services internes hébergés sur un réseau privé à partir d'un environnement externe.[3]
  Fonctionnement
  - Un tunnel est établi entre un hôte local et un serveur distant.
  - Les requêtes envoyées vers un port sur le serveur distant sont redirigées vers un port de l'hôte local.
  - Les utilisateurs distants accèdent au service comme s'il était hébergé sur le serveur distant.

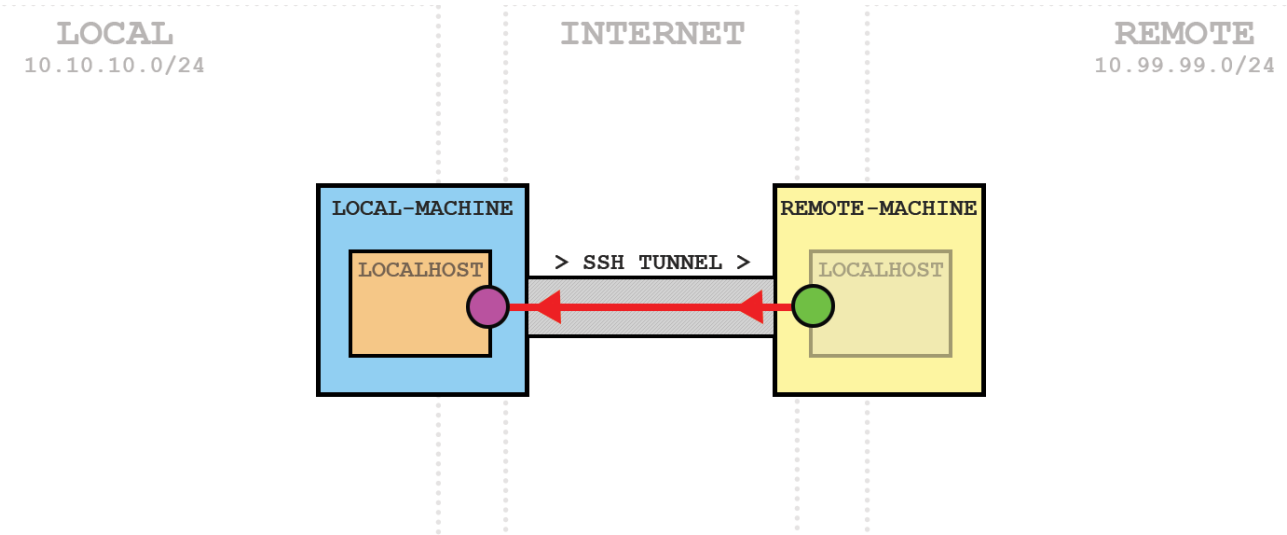


FIGURE 1.3 : Présentation d'une architecture basée sur la redirection de port à distance [7]

- **Redirection de port dynamique (Dynamic port forwarding)** : Le port forwarding dynamique est une méthode plus flexible, qui transforme un hôte local en un proxy SOCKS. Contrairement aux types local et distant, le port forwarding dynamique ne cible pas un service ou un port spécifique mais permet de rediriger tout type de trafic réseau vers diverses destinations à travers un seul port local.[3]

  Fonctionnement

- L'utilisateur configure un tunnel via SSH pour établir un proxy SOCKS.
- Les applications (comme un navigateur) sont configurées pour utiliser ce proxy.
- Le proxy redirige le trafic en fonction des requêtes, offrant un accès sécurisé à divers services.

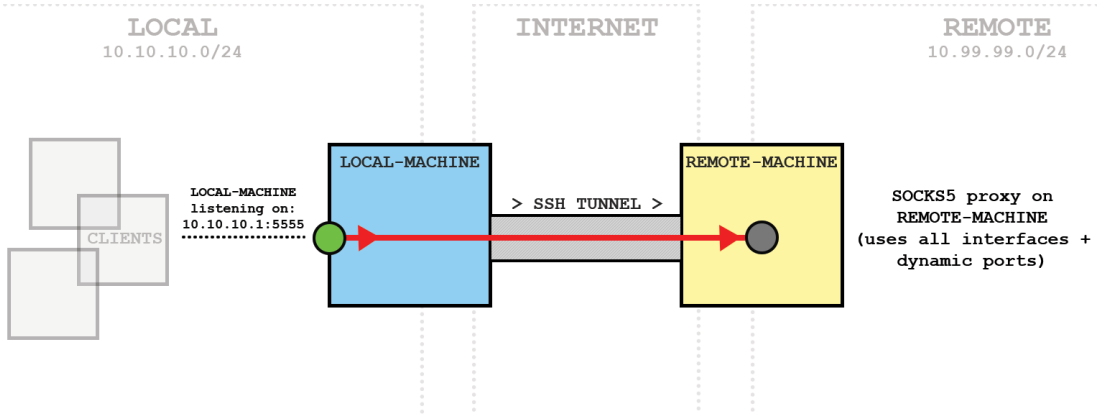


FIGURE 1.4 : Présentation d'une architecture basée sur la redirection de port dynamique [7]

Tableau récapitulatif des différents types de port forwarding

Types	Destination	Usage principal	Exemple typique
Local	De local à distant	Accéder à des services distants	Connexion à une base de données distante
Distant	De distant à local	Partager des services locaux	Héberger un site local accessible à l'extérieur
Dynamique	Vers plusieurs hôtes	Proxy sécurisé pour divers usages	Navigation sécurisée ou accès contourné

TABLE 1.1 : Comparaison des types de port forwarding

1.2.3 Différences et similitudes avec d'autres techniques de tunneling

Le port forwarding, bien qu'étant une forme de tunneling, se distingue par ses spécificités et ses usages. Comparé à d'autres techniques de tunneling telles que le VPN, IPSec ou SSL/TLS, il offre une approche ciblée et simplifiée pour rediriger le trafic réseau. Cependant, il partage également certains points communs avec ces méthodes, notamment en matière de sécurité et d'encapsulation des données.

Différences principales

- Le port forwarding se concentre sur la redirection de trafic vers des ports ou services spécifiques, tandis que des technologies comme le VPN encapsulent l'ensemble du trafic réseau entre deux points.
- Contrairement au VPN, le port forwarding ne nécessite pas toujours une connexion persistante, ce qui peut en faire une solution plus légère pour certains cas d'usage.
- Les protocoles comme SSL/TLS offrent une sécurité généralisée pour toutes les communications sur une session, tandis que le port forwarding est souvent limité à un ou plusieurs ports définis.

Similitudes clés

- Tous ces mécanismes assurent la sécurité des données transmises grâce à des méthodes de chiffrement.
- Ils permettent de contourner les restrictions réseau en encapsulant les paquets pour les rendre illisibles aux systèmes intermédiaires.

1.3 Avantages et limites du port forwarding

1.3.1 Sécurisation des communications réseau

Le port forwarding joue un rôle crucial dans la sécurisation des communications réseau, en permettant un contrôle précis et ciblé du trafic réseau. Cette méthode repose souvent sur des protocoles sécurisés tels que SSH ou SSL/TLS, qui chiffrent les données en transit et empêchent leur interception ou modification par des tiers.

Chiffrement des données : Lorsqu'il est configuré avec des protocoles comme SSH, le port forwarding encapsule et chiffre le trafic, garantissant que les données transmises entre l'émetteur et le destinataire restent confidentielles. Cela le rend particulièrement adapté pour des communications sur des réseaux non sécurisés, tels que les points d'accès Wi-Fi publics.

Réduction des surfaces d'attaque : En limitant l'exposition des services internes à des ports spécifiques et à des utilisateurs autorisés, le port forwarding réduit les surfaces d'attaque potentielles. Seuls les ports configurés pour la redirection sont ouverts aux connexions externes, et toutes les autres tentatives sont rejetées par le pare-feu.

Authentification renforcée : L'intégration d'un mécanisme d'authentification, tel que l'utilisation de clés SSH ou de certificats SSL/TLS, améliore encore la sécurité en garantissant que seules les parties autorisées peuvent établir une connexion. Cela prévient les accès non autorisés et les attaques par force brute.

1.3.2 Optimisation des performances réseau

Le port forwarding contribue également à l'optimisation des performances réseau, en permettant une gestion ciblée et efficace du trafic. En redirigeant uniquement les données nécessaires à des destinations précises, cette méthode réduit la charge globale sur les ressources réseau, tout en améliorant la fluidité des communications.

Réduction du trafic inutile : Contrairement à d'autres techniques comme le VPN, qui encapsule l'ensemble du trafic entre deux points, le port forwarding se limite à des ports ou services spécifiques. Cela réduit la quantité de données transmises, optimisant ainsi l'utilisation de la bande passante et minimisant les congestions réseau.

Accès direct aux services : En fournissant un accès direct à des services ou applications via des

ports redirigés, le port forwarding élimine les étapes intermédiaires, ce qui réduit la latence. Par exemple, un serveur web interne peut être accessible directement par une redirection de port, évitant des routes complexes.

Efficacité dans les environnements distribués : Dans des infrastructures réparties, telles que les environnements cloud ou les réseaux hybrides, le port forwarding permet de concentrer le trafic réseau sur des points spécifiques. Cela facilite la gestion des connexions entrantes et sortantes, tout en optimisant l'allocation des ressources.

Flexibilité et adaptabilité : La redirection de port dynamique (Dynamic Port Forwarding) offre une flexibilité supplémentaire en agissant comme un proxy SOCKS. Cela permet de rediriger le trafic vers plusieurs destinations tout en utilisant une seule configuration, ce qui est particulièrement utile dans des scénarios où les besoins en connectivité évoluent rapidement.

1.3.3 Limitations et défis associés

Bien que le port forwarding présente de nombreux avantages, il n'est pas exempt de limitations et de défis. Ces contraintes peuvent varier en fonction du contexte d'utilisation, des configurations réseau et des exigences de sécurité.

Complexité de la configuration : Le port forwarding nécessite une configuration manuelle, notamment sur les routeurs, pare-feu ou serveurs intermédiaires. Cette tâche peut être difficile pour les utilisateurs non techniques, surtout lorsqu'il s'agit de définir des règles précises pour la redirection des ports.

Vulnérabilité aux attaques si mal configuré : Une configuration incorrecte du port forwarding peut exposer des ports sensibles à des attaques externes, telles que :

- Les attaques par force brute visant à accéder à des services redirigés.
- Les scans de ports, qui peuvent révéler la présence de ports ouverts aux attaquants.

Sans des mécanismes de sécurité appropriés, comme le filtrage IP ou l'authentification renforcée, cette technique peut devenir un vecteur de menaces.

Absence de chiffrement natif : Contrairement aux technologies comme les VPN ou SSL/TLS, le port forwarding ne chiffre pas les données par défaut. Lorsqu'il est utilisé seul, il peut exposer les informations sensibles en transit. Pour garantir la sécurité, il est souvent nécessaire de le coupler à un protocole de tunneling sécurisé comme SSH.

Scalabilité limitée : Dans les environnements complexes ou de grande envergure, où plusieurs services ou ports doivent être redirigés, le port forwarding peut devenir difficile à gérer. La multiplication des règles de redirection alourdit la maintenance et augmente les risques d'erreurs.

Non adapté pour le trafic global : Contrairement à un VPN qui encapsule tout le trafic entre deux points, le port forwarding est limité à des ports ou services spécifiques. Cette portée restreinte peut être un obstacle dans les scénarios où une protection ou une redirection globale est requise.

Conclusion

En conclusion, ce chapitre a exploré les fondements du tunneling et du port forwarding, en mettant en lumière leurs mécanismes, leurs applications et leurs avantages. Les différents types de port forwarding ainsi que leurs spécificités ont été détaillés, tout en soulignant leurs limites. Cette analyse théorique fournit une base solide pour la mise en œuvre pratique de solutions basées sur le port forwarding dans les chapitres suivants.

Chapitre 2

Automatisation du port forwarding

Introduction

Ce chapitre explore le port forwarding comme une application spécifique des techniques de tunneling, visant à optimiser les communications réseau. Une analyse approfondie de son fonctionnement et de ses avantages pratiques sera proposée. L'accent sera mis sur les outils nécessaires à l'implémentation d'un script automatisé, ainsi que sur les configurations essentielles pour garantir à la fois l'efficacité et la sécurité des connexions.

2.1 Présentation de la redirection de port

2.1.1 Fonctionnement du port forwarding dans le tunneling

Le port forwarding est une technique clé dans le domaine du tunneling, permettant d'accéder de manière sécurisée et ciblée à des services ou ressources spécifiques hébergés sur un réseau privé. Cette méthode repose sur la redirection du trafic depuis un port public vers une adresse IP et un port privés, souvent à l'aide de protocoles comme SSH. Elle optimise les communications réseau en simplifiant les configurations tout en renforçant la sécurité et la flexibilité des accès.

Le fonctionnement du port forwarding dans ce contexte suit trois étapes principales :

Etape 1 : Réception du paquet entrant
Lorsqu'un utilisateur externe envoie une requête vers un service spécifique via un port défini (sur un routeur, un pare-feu ou une machine intermédiaire), cette requête est interceptée.

Etape 2 : Réécriture de l'adresse de destination
Le dispositif intermédiaire redirige la requête en modifiant l'adresse IP de destination et le port. Cette étape garantit que le trafic est transmis au service ou à l'appareil interne approprié.

Etape 3 : Transmission et réponse
Le paquet est acheminé vers sa destination finale, permettant à l'utilisateur d'interagir avec le service comme s'il se trouvait sur le même réseau local. Les réponses suivent un chemin inverse pour établir une communication fluide et bidirectionnelle.

Grâce à cette approche, le port forwarding permet un accès sécurisé aux ressources internes, tout en encapsulant et en chiffrant les échanges via des tunnels. Cette technique constitue une solution efficace pour optimiser les communications réseau, qu'il s'agisse de sécuriser des connexions dans des environnements distribués ou de simplifier l'accès à des services critiques.

2.2 Avantages du port forwarding pour les activités des petites entreprises et startups

Le port forwarding, en tant qu'application spécifique des techniques de tunneling, offre plusieurs avantages significatifs pour les startups et les petites entreprises. Cette méthode permet de centraliser les ressources au sein du siège principal tout en offrant un accès sécurisé et optimisé aux succursales régionales ou aux collaborateurs en déplacement.

Centralisation et coordination des activités : Grâce au port forwarding, les succursales et bureaux décentralisés peuvent accéder en toute sécurité aux services et ressources centralisés. Cela favorise une meilleure coordination des opérations, tout en simplifiant la gestion des systèmes internes.

Mobilité des employés : Pour les employés en déplacement, le port forwarding offre une solution efficace pour accéder aux systèmes internes de l'organisation. Cette continuité d'accès garantit que les activités ne sont pas interrompues, même lorsque les collaborateurs travaillent en dehors des locaux.

Réduction des coûts d'infrastructure : En fournissant une alternative légère à l'installation d'un VPN complet, le port forwarding permet aux entreprises de réduire les coûts liés à l'infrastructure. En utilisant des tunnels sécurisés, tels que ceux basés sur SSH, cette solution garantit un niveau de sécurité élevé tout en restant simple à déployer.

Sécurité ciblée et gestion simplifiée : Le port forwarding offre un accès spécifique à des services internes sans exposer l'ensemble du réseau. Cela simplifie la gestion des ressources et réduit les risques d'exposition aux menaces extérieures. Les tunnels sécurisés protègent les échanges tout en permettant une administration centralisée et flexible.

Soutien aux formations et collaborations : En facilitant l'accès sécurisé aux ressources internes, le port forwarding contribue à soutenir les besoins en formation continue et les collaborations avec des partenaires ou des clients.

2.3 Implémentation du script pour la redirection de port

2.3.1 Fonctionnalité du script

Le script proposé permet d'automatiser la configuration de deux types de redirection de port : local et dynamique, apportant ainsi une flexibilité accrue dans l'accès aux services distants. Il proposera d'abord à l'utilisateur de configurer une paire de clés SSH pour sécuriser les connexions, puis de choisir parmi les deux options de redirection. La redirection de port local (L) permet à une machine locale d'accéder à un service spécifique distant via une machine intermédiaire. Enfin, la redirection de port dynamique (D) utilise un proxy SOCKS, permettant de relayer le trafic réseau vers plusieurs services, contrairement à la redirection de port local via la machine intermédiaire. Ce script simplifie ainsi considérablement la gestion des tunnels SSH pour chaque type de redirection.

2.3.2 Outils choisis pour l'implémentation

La redirection de port peut être implémentée à l'aide de divers logiciels, outils ou protocoles, chacun ayant ses spécificités. Parmi les solutions couramment utilisées, on peut citer Ngrok, Metasploit, Netcat, Plink, ainsi que le protocole SSH. Chacune de ces options offre des fonctionnalités variées en matière de mise en place de tunnels pour le transfert de données entre différents réseaux. Toutefois, dans le cadre de notre projet, le choix s'est porté sur le protocole SSH pour son aspect sécurisé, notamment grâce à son chiffrement robuste des communications. Pour faciliter la mise en œuvre, un script en BASH sera utilisé afin d'automatiser le processus de redirection de port. Ce script permettra de gérer efficacement les connexions entre les sites, tout en garantissant la confidentialité et l'intégrité des données échangées grâce au protocole SSH.

2.3.3 Configuration SSH adoptée pour sécuriser les connexions

SSH (Secure Shell) est un protocole de communication réseau qui permet d'établir des connexions sécurisées entre deux machines. Il utilise un chiffrement robuste pour protéger les données échangées, garantissant ainsi la confidentialité et l'intégrité des informations. Dans le contexte d'une solution de redirection de port, le protocole SSH joue un rôle central en offrant une couche de sécurité pour le transfert de données.[6]

Pour renforcer la sécurité des connexions SSH dans le contexte de la redirection de port, il est essentiel d'appliquer des configurations qui limitent l'accès et imposent des méthodes d'authentification robustes. Voici deux configurations spécifiques pour cette mise en place :

- **Désactivation de l'accès shell** : Afin d'éviter que les utilisateurs puissent accéder au terminal du serveur via SSH, il est possible de restreindre l'accès au shell avec l'option ForceCommand dans le fichier de configuration SSH (/etc/ssh/sshd_config). En configurant cette option, les utilisateurs peuvent être redirigés automatiquement vers une commande spécifique ou, si nécessaire, recevoir un message de non-disponibilité du terminal.

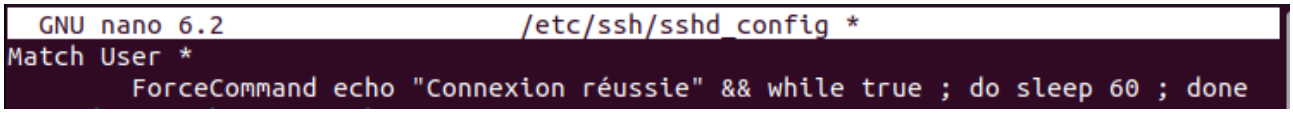


FIGURE 2.1 : Configuration du fichier sshd_config pour la désactivation de l'accès shell

Cette configuration permet d'utiliser SSH uniquement pour la redirection de port, sans fournir d'accès interactif au shell du serveur.

- **Authentification par clé publique/privée** : La mise en place de l'authentification par clé SSH améliore la sécurité en imposant une connexion par clés plutôt que par mots de passe. Pour activer cette méthode d'authentification, on s'assure que les lignes suivantes sont configurées dans /etc/ssh/sshd_config :

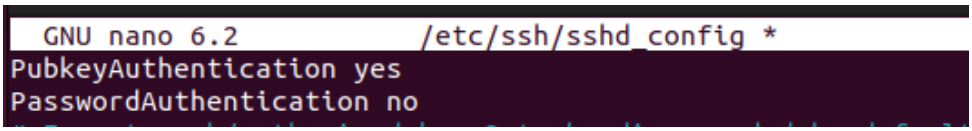


FIGURE 2.2 : Configuration du fichier sshd_config pour l'activation de l'authentification par clé SSH uniquement

Les utilisateurs devront générer une paire de clés avec ssh-keygen (sur leur machine locale) et placer la clé publique dans le fichier /.ssh/authorized_keys sur le serveur SSH. Cela assure une authentification plus sûre et prévient les attaques basées sur le vol de mots de passe.

NB : Suite à la modification du fichier de configuration ssh (/etc/ssh/sshd_config) il est impérativement nécessaire de redémarrer le service ssh grâce à cette commande :

```
$ sudo systemctl restart sshd
```

2.4 Mise en place du script de redirection

Ce script Bash a pour objectif d'automatiser la configuration de redirections de port SSH. Il commence par demander à l'utilisateur s'il souhaite générer une paire de clés SSH pour sécuriser les connexions. Si l'utilisateur choisit de configurer les clés, celles-ci seront créées dans un chemin spécifié, avec la possibilité d'ajouter une passphrase pour plus de sécurité. Si aucune configuration n'est nécessaire, le script passe à la suite.

```
# Demande le type de redirection de port
echo "Choisissez le type de port forwarding :"
echo "1) Local Port Forwarding"
echo "2) Dynamic Port Forwarding"
read -p "Entrez le numéro correspondant : " PORT_FORWARDING_CHOICE

# Demande l'IP publique de la machine intermédiaire
read -p "Veuillez entrer l'IP de la machine intermédiaire : " INTERMEDIATE_IP

# Demande le nom d'utilisateur de la machine intermédiaire
read -p "Veuillez entrer le nom d'utilisateur de la machine intermédiaire : " USERNAME
```

FIGURE 2.4 : Choix de la redirection de port

```
#!/bin/bash

# Message de bienvenue
echo "Bienvenue dans le processus d'automatisation de la redirection de port."
# Demande à l'utilisateur s'il souhaite configurer une paire de clés SSH
read -p "Voulez-vous configurer une paire de clés SSH ? (o/n) " CONFIGURE_KEYS
# Vérification et création des clés SSH si nécessaire
if [[ "$CONFIGURE_KEYS" =~ ^[oO]$ ]]; then
    KEY_PATH="$HOME/.ssh/id_rsa"

    echo "Les clés SSH seront enregistrées dans $KEY_PATH."
    # Création de la paire de clés
    read -p "Voulez-vous ajouter une passphrase à votre clé SSH ? (o/n) " RESPONSE
    if [[ "$RESPONSE" =~ ^[oO]$ ]]; then
        ssh-keygen -t rsa -b 2048 -f "$KEY_PATH"
    else
        ssh-keygen -t rsa -b 2048 -f "$KEY_PATH" -N ""
    fi
        echo "Clé SSH générée avec succès."
    echo "N'oubliez pas d'ajouter la clé publique à 'authorized_keys' sur la machine intermédiaire."
else
    echo "Aucune configuration de clé SSH n'est nécessaire, poursuite du processus."
fi

# Demande à l'utilisateur de choisir une clé
read -p "Entrez le chemin de la clé privée à utiliser (ou appuyez sur Entrée pour utiliser $KEY_PATH) : " USER_KEY_PATH
USER_KEY_PATH=${USER_KEY_PATH:-$KEY_PATH}  # Utilise la clé par défaut si aucun chemin n'est fourni
```

FIGURE 2.3 : Création de paires clés SSH

Ensuite, le script demande à l'utilisateur de sélectionner la clé privée à utiliser pour l'authentification, puis le type de redirection de port à configurer : soit le Local Port Forwarding, soit le Dynamic Port Forwarding. L'utilisateur est ensuite invité à entrer l'adresse IP et le nom d'utilisateur de la machine intermédiaire pour finaliser la configuration. Ces informations sont essentielles pour établir une connexion sécurisée avec la machine intermédiaire en vue de réaliser la redirection de port spécifiée.

2.4.1 Redirection de port local

La redirection locale permet de transférer un port spécifique de la machine locale vers un port spécifique du serveur distant. Cela peut être utile lorsqu'un service du serveur distant n'est pas accessible depuis la machine locale.[9] La syntaxe de la redirection locale est la suivante :

```
$ ssh -L local_port:remote_host:remote_port user@remote_server
```

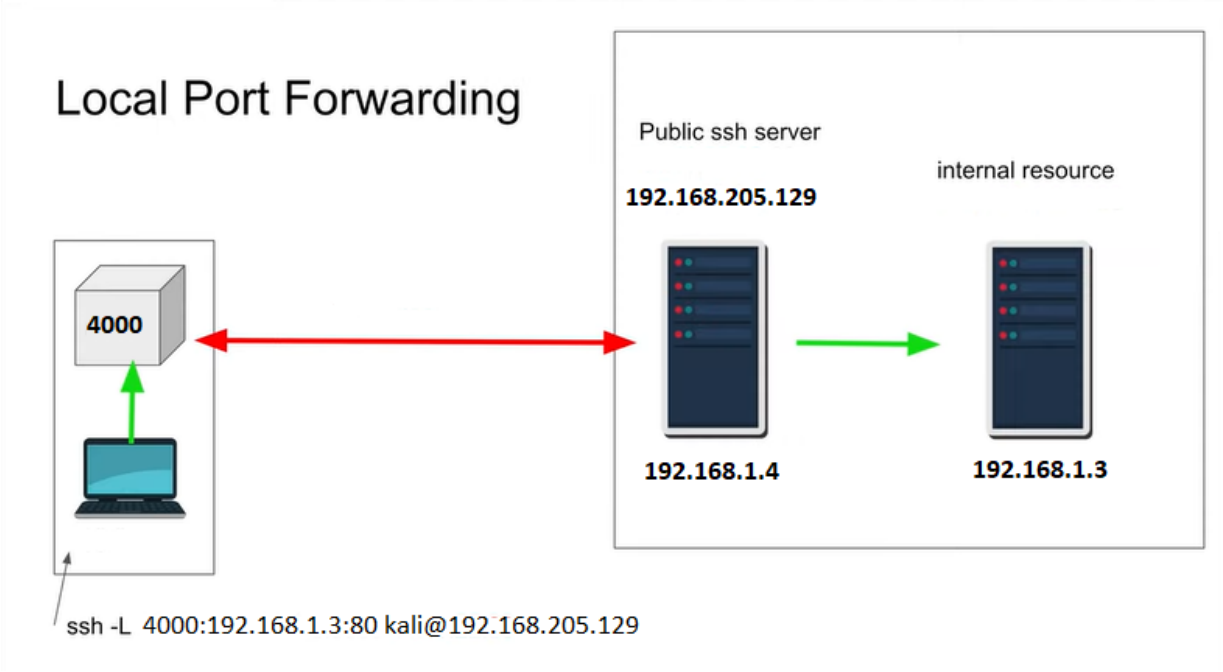


FIGURE 2.5 : Illustration de la redirction locale avec le protocole SSH [9]

Dans la partie de la redirection locale du script, l'utilisateur est guidé pour fournir les informations nécessaires à la mise en place de la redirection de port. Le script demande de définir le port local, soit en spécifiant un port personnalisé, soit en choisissant un port aléatoire dans la plage de 9000 à 65535. Si un port personnalisé est choisi, le script valide la saisie de l'utilisateur pour s'assurer qu'il est dans la plage appropriée. Par la suite, l'utilisateur est invité à entrer l'adresse IP de la machine sur le réseau intranet qu'il veut atteindre, ainsi que le port distant sur lequel le service est actif. Enfin, le script lance la redirection locale en exécutant la commande SSH approprié déjà défini.

```
# Gestion des choix de port forwarding
case $PORT_FORWARDING_CHOICE in
    1)
        # Redirection de port local
        echo "Vous avez choisi la redirection de port local."
        echo "Choisissez comment définir le port local :"
        echo "1) Port local personnalisé"
        echo "2) Port aléatoire (entre 9000 et 65535)"
        read -p "Entrez le numéro correspondant : " PORT_CHOICE
        if [ "$PORT_CHOICE" == "1" ]; then
            read -p "Veuillez entrer le port local : " LOCAL_PORT
            # Vérification de la validité du port
            if [[ $LOCAL_PORT -lt 9000 || $LOCAL_PORT -gt 65535 ]]; then
                echo "Port local non valide. Veuillez entrer un port entre 9000 et 65535."
                exit 1
            fi
        elif [ "$PORT_CHOICE" == "2" ]; then
            LOCAL_PORT=$((RANDOM % 56535 + 9000))  # Port aléatoire entre 9000 et 65535
            echo "Un port local aléatoire a été sélectionné : $LOCAL_PORT"
        else
            echo "Choix invalide. Veuillez relancer le script."
            exit 1
        fi
        # Demande l'IP de la machine cible du réseau intranet
        read -p "Veuillez entrer l'IP de la machine cible du réseau intranet : " TARGET_IP
        # Demande le port distant sur lequel tourne le service sur la machine cible
        read -p "Veuillez entrer le port distant sur lequel le service tourne sur la machine cible : " TARGET_PORT
        echo "Lancement de la redirection de port local : ssh -i $USER_KEY_PATH -L $LOCAL_PORT:$TARGET_IP:$TARGET_PORT $USERNAME@$INTERMEDIATE_IP"
        ssh -i "$USER_KEY_PATH" -L "$LOCAL_PORT:$TARGET_IP:$TARGET_PORT" "$USERNAME@$INTERMEDIATE_IP"
        ;;
```

FIGURE 2.6 : Script de la redirection de port local

Cette alternative permet de rendre disponible un service spécifique, avec un script exécuté côté client qui se connecte à un serveur faisant office de passerelle entre le site distant (Parakou) et le réseau local.

2.4.2 Redirection de port dynamique

La Redirection de port dynamique est une technique de tunneling plus flexible qui permet de transformer une machine en un Proxy SOCKS. Contrairement aux redirections de port local et distant, cette méthode ne se limite pas à une seule destination. Elle peut gérer plusieurs connexions et rediriger le trafic vers plusieurs hôtes ou ports, en fonction des demandes envoyées via le proxy. En d'autres termes, avec la redirection dynamique, un utilisateur peut accéder à différents services sur un réseau distant via un seul tunnel sécurisé. La syntaxe de la redirection dynamique est la suivante :

```
$ ssh -D local_port user@remote_server
```

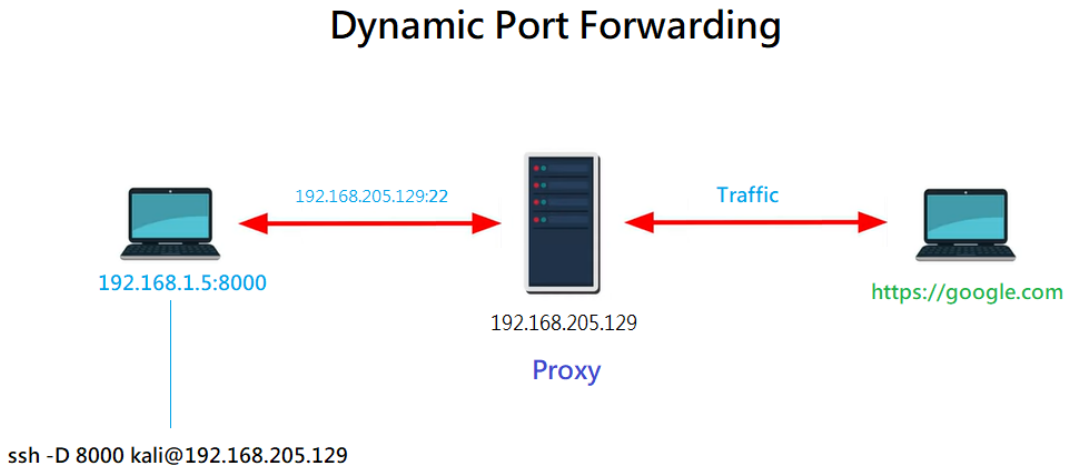


FIGURE 2.7 : Illustration de la redirection de port dynamique avec le protocole SSH [9]

Cette partie du script présente les informations à entrer par l'utilisateur pour la redirection dynamique. Après avoir choisi le port, la commande SSH est exécutée pour établir la redirection de port dynamique, permettant ainsi à l'utilisateur de se connecter au serveur intermédiaire via un tunnel SOCKS, offrant une méthode flexible pour accéder aux services réseau.

```
2)
    # Redirection de port dynamique
    echo "Vous avez choisi la redirection de port dynamique."

    # Redirection de port distant
    echo "Vous avez choisi la redirection de port dynamique."
    echo "Choisissez comment définir le port dynamique :"
    echo "1) Port dynamique personnalisé"
    echo "2) Port aléatoire (entre 9000 et 65535)"
    read -p "Entrez le numéro correspondant : " PORT_CHOICE

    if [ "$PORT_CHOICE" == "1" ]; then
        read -p "Veuillez entrer le port dynamique : " REMOTE_PORT
        # Vérification de la validité du port
        if [[ $REMOTE_PORT -lt 9000 || $REMOTE_PORT -gt 65535 ]]; then
            echo "Port dynamique non valide. Veuillez entrer un port valide."
            exit 1
        fi
    elif [ "$PORT_CHOICE" == "2" ]; then
        REMOTE_PORT=$((RANDOM % 56535 + 9000))  # Port aléatoire entre 9000 et 65535
        echo "Un port dynamique aléatoire a été sélectionné : $REMOTE_PORT"
    else
        echo "Choix invalide. Veuillez relancer le script."
        exit 1
    fi

    echo "Lancement de la redirection de port dynamique : ssh -i $USER_KEY_PATH -D $REMOTE_PORT $USERNAME@$INTERMEDIATE_IP"
    ssh -i "$USER_KEY_PATH" -D "$REMOTE_PORT" "$USERNAME@$INTERMEDIATE_IP"
    ;;
```

FIGURE 2.8 : Script de redirection de port dynamique

La redirection de port dynamique est particulièrement avantageuse, car elle offre un accès sécurisé à plusieurs services internes via un seul tunnel, sans nécessiter la création de redirections individuelles pour chaque port. À ce stade, le script s'exécute côté client.

Enfin, pour gérer les mauvaises entrées de l'utilisateur, ce code vérifie si une option invalide a été sélectionnée. Dans ce cas, le script affiche le message "Choix invalide. Veuillez relancer le script." pour informer l'utilisateur de l'erreur, puis met fin à l'exécution avec la commande exit 1, signalant ainsi un arrêt en raison d'une erreur. Cette validation permet de s'assurer que seules des options correctes sont acceptées, renforçant la fiabilité et la robustesse du script.

```
*)
    echo "Choix invalide. Veuillez relancer le script."
    exit 1
    ;;
```

FIGURE 2.9 : Enter Caption

2.4.3 Redirection de port distant

La redirection de port distant inverse le processus de la redirection de port local. Il est important de noter que, tout comme la redirection de port local, la redirection de port distant permet de rendre accessible un service spécifique. Dans le cas de la redirection de port local, un utilisateur accède à un service distant depuis sa machine locale. À l'inverse, avec la redirection de port distant, un service local est rendu accessible à des utilisateurs distants en passant par un serveur intermédiaire, ce qui étend l'accessibilité du service au-delà du réseau local. Ce type de redirection est configuré sur la machine intermédiaire ou le serveur à atteindre depuis l'extérieur du réseau. Bien qu'il permette d'atteindre le même objectif que la redirection de port local, il n'a pas été implémenté dans ce projet.La syntaxe de la redirection de port distante est la suivante :

```
$ ssh -R remote_port:remote_host:local_port user@remote_server
```

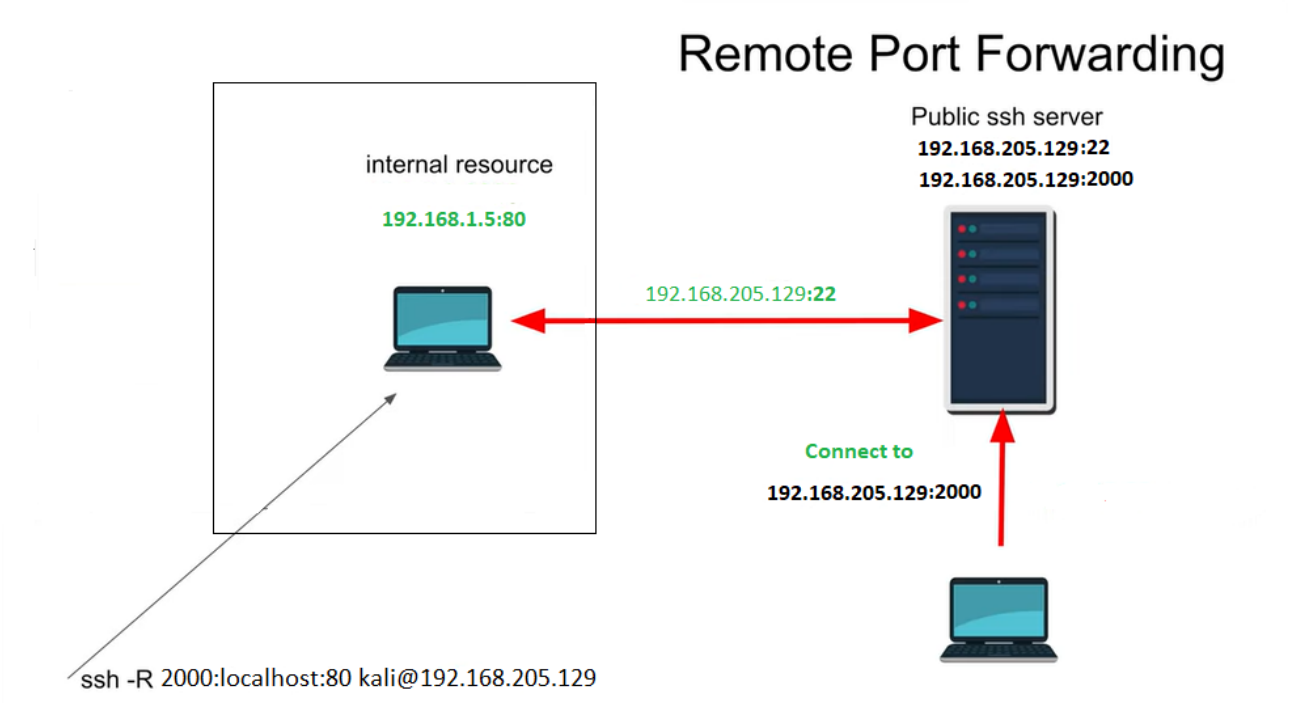


FIGURE 2.10 : Illustration de la redirection de port distant [9]

Le script complet (**Automator_port_forwarding.sh**) voir en Annexe 3.1.2.

Conclusion

Ce chapitre a présenté la conception et l'implémentation d'un script pour automatiser le port forwarding, en explorant les techniques locales, dynamiques et distantes. Cette solution, basée sur SSH, offre une méthode flexible et sécurisée pour optimiser les communications réseau, tout en simplifiant leur gestion. Elle constitue une base pratique pour des applications concrètes dans des environnements variés.

Chapitre 3

Test et fonctionnement du script de redirection de port

Introduction

Dans ce chapitre, nous mettrons l'accent sur l'évaluation du fonctionnement du script de redirection de port développé dans le chapitre précédent. L'objectif est de présenter des cas d'utilisation concrets de la redirection de port dans le cadre des activités d'une startup. Cette analyse visera à démontrer l'efficacité du script, à illustrer comment il répond aux besoins spécifiques de la structure et à explorer les perspectives d'utilisation futures.

3.1 Test et fonctionnement du script

Ce script, nommé **Automator_port_forwarding.sh**, permet de configurer la redirection de port local et dynamique en utilisant une connexion SSH via une machine intermédiaire. Cette section se concentrera sur les différentes options de redirection de port afin d'illustrer les fonctionnalités du script. Avant l'utilisation de ce script, il est nécessaire de lui attribuer les droits d'exécution en utilisant la syntaxe suivante :

```
$ chmod +x Automator_port_forwarding.sh
```

Avant de poursuivre, il convient de rappeler que les configurations SSH, ainsi que la génération des clés, ont déjà été effectuées. Dans notre simulation, les adresses IP de la plage 192.168.122.0/24 seront considérées comme des adresses publiques, tandis que celles de la plage 10.0.8.0/24 seront considérées comme privées. La simulation implique une machine physique (Kali) et deux machines virtuelles (server_VPN et client_VPN). La machine physique communique via le réseau public (192.168.122.0/24) avec la machine virtuelle (client_VPN). La machine physique (Kali), représentant la machine intermédiaire, communique avec la machine virtuelle (server_VPN) sur le réseau privé (10.0.8.0/24). Les images suivantes montrent les configurations réseau entre les machines :

Présentation de la configuration IP de la machine virtuelle cliente (client_VPN)

```
client_vpn@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 52:54:00:d8:56:5e brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.167/24 brd 192.168.122.255 scope global dynamic noprefixrou
te enp1s0
       valid_lft 2666sec preferred_lft 2666sec
    inet6 fe80::2964:a851:ca7c:dbc5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
client_vpn@ubuntu:~$
```

FIGURE 3.1 : Configuration réseau de la machine cliente (client_VPN)

Présentation de la configuration IP de la machine intermédiaire (Kali)

```
┌──(kali㉿dias)-[~/Documents/Mémoire_IFRI/scripts]
└─$ ifconfig | grep -A 8 -E 'virbr0|virbr2'
virbr0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.122.1  netmask 255.255.255.0  broadcast 192.168.122.255
        ether 52:54:00:9c:6c:13  txqueuelen 1000  (Ethernet)
        RX packets 75542  bytes 4711023 (4.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 208643  bytes 290070387 (276.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

virbr1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
--
virbr2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.8.1  netmask 255.255.255.0  broadcast 10.0.8.255
        ether 52:54:00:37:d1:81  txqueuelen 1000  (Ethernet)
        RX packets 47380  bytes 3267251 (3.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 261919  bytes 365903310 (348.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

FIGURE 3.2 : Configuration réseau de la machine intermédiaire (Kali)

Présentation de la configuration IP de la machine virtuelle à atteindre (server_VPN)

```
server_vpn@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 52:54:00:9b:da:5a brd ff:ff:ff:ff:ff:ff
    inet 10.0.8.214/24 brd 10.0.8.255 scope global dynamic noprefixroute enp1s0
       valid_lft 3377sec preferred_lft 3377sec
    inet6 fe80::2b91:573c:1876:ad0b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOW
N group default
    link/ether 02:42:e3:7b:50:54 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
server_vpn@ubuntu:~$
```

FIGURE 3.3 : Configuration réseau de la machine virtuelle à atteindre (server_VPN)

3.1.1 Exemple d'utilisation de la redirection de port local

Dans cette présentation, un serveur FTP est actif sur la machine virtuelle server (server_VPN, IP : 10.0.8.214), accessible via le réseau. Ce serveur permettra à l'utilisateur distant d'avoir accès à des fichiers distants et de télécharger des fichiers sur le serveur à travers un tunnel sécurisé grâce au protocole SSH.
Teste de connectivité au serveur FTP déployé sur le serveur de la machine server (server_VPN) depuis la machine intermédiaire.

```
┌──(kali㉿dias)-[~/Documents/Mémoire_IFRI/scripts]
└─$ ftp 10.0.8.214
Connected to 10.0.8.214.
220 (vsFTPd 3.0.5)
Name (10.0.8.214:kali): server_vpn
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```

FIGURE 3.4 : Test de connexion au serveur FTP depuis la machine intermédiaire

Ensuite, on démarre sur la machine cliente (client_VPN) le script **Automator_port_forwarding.sh** : en spécifiant le chemin absolu pour atteindre la clé privée qui nous servira dans la suite du programme.

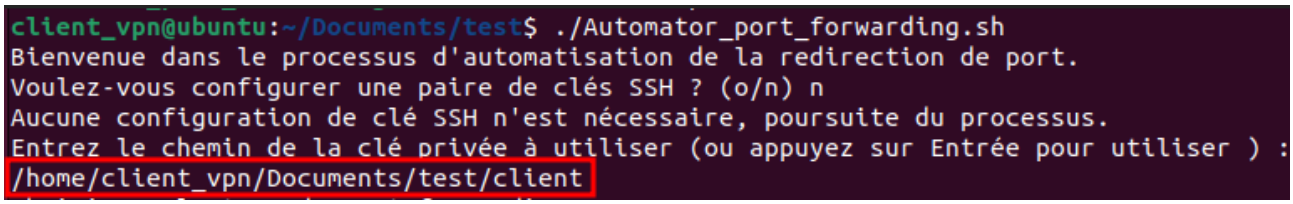


FIGURE 3.5 : Lancement du script Automator_port_forwarding.sh

Dans la suite, le programme nous permet de choisir le type de redirection à mettre en place, et ensuite de fournir l'adresse IP et le nom d'utilisateur de la machine intermédiaire.

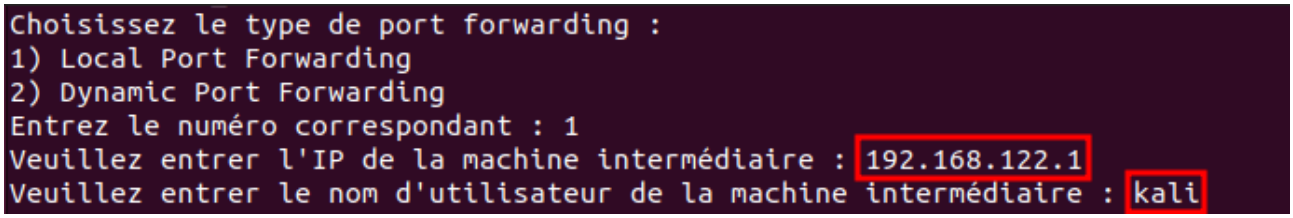


FIGURE 3.6 : Choix de la redirection de port local

Après, le choix de la redirection de port local, le programme nous permet d'entrer le port de sortie des requêtes de la machine cliente vers le serveur. Ensuite, on entre l'adresse IP du serveur (server_VPN) dans le réseau intranet et le port du service à atteindre (dans notre cas, il s'agit du port 21 pour le service FTP). Après cela, il démarre la redirection de port à travers la commande :

```
$ ssh -i /home/client_vpn/Documents/test/client -L
   9002:10.0.8.214:21 kali@192.168.122.1
```



FIGURE 3.7 : Lancement de la redirection de port local

Ainsi, pour permettre à l'utilisateur distant d'avoir accès au serveur FTP depuis sa machine grâce à la redirection de port, il doit entrer cette commande :

```
$ ftp localhost 9002
```

```
client_vpn@ubuntu:~/Documents/test$ ftp localhost 9002
Connected to localhost.
220 (vsFTPd 3.0.5)
Name (localhost:client_vpn): server_vpn
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

FIGURE 3.8 : Connexion de la machine cliente au serveur FTP

Grâce à l'outil d'analyse réseau **Wireshark**, il est possible d'observer que la communication entre l'intermédiaire et le client est chiffrée. Cette partie de la communication doit en effet être chiffrée, car elle transite par le réseau public (Internet).

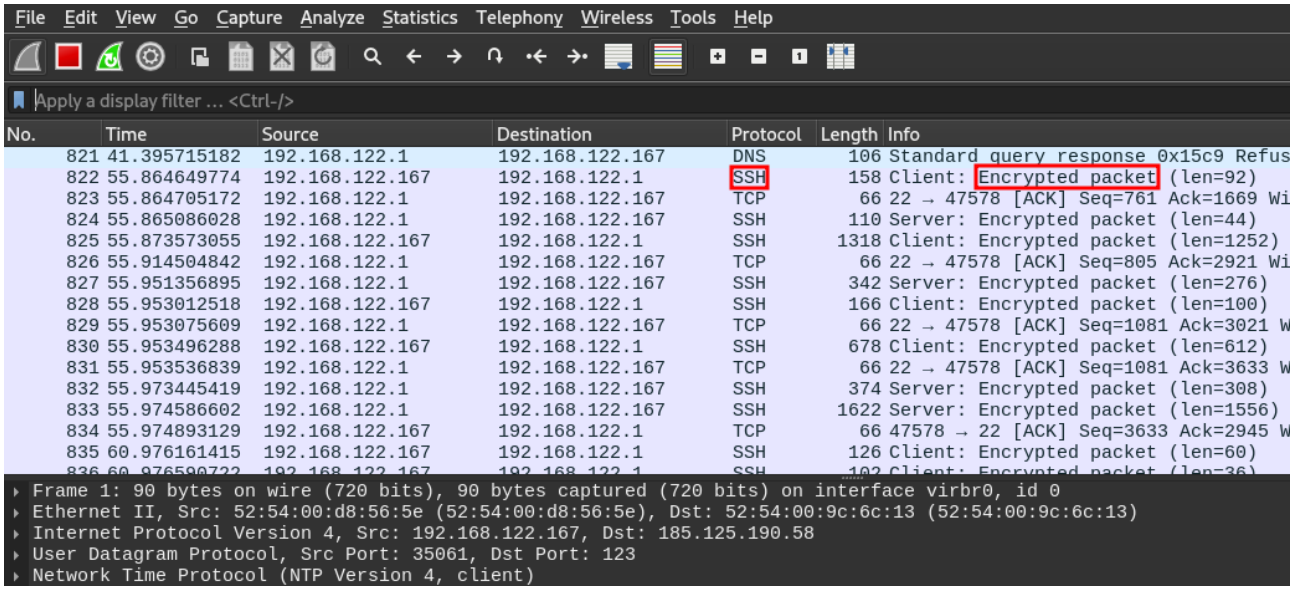


FIGURE 3.9 : Analyse réseau à travers l'outil Wireshark

3.1.2 Exemple d'utilisation de la redirection de port dynamique

Pour la présentation de la redirection de port dynamique, une page login et un conteneur docker **WordPress** ont été déployés sur la machine server_VPN.

```
server_vpn@ubuntu:~$ sudo docker run --name wordpress --link wordpress-db:mysql -p 8080:80
-e WORDPRESS_DB_HOST=wordpress-db:3306 -e WORDPRESS_DB_USER=[redacted] -e WORDPRESS_DB_P
ASSWORD=[redacted] -e WORDPRESS_DB_NAME=[redacted] -d wordpress
7b50bf2d8f4191dbe9ac1385cd8[redacted]604e0b995cc09c20bc
server_vpn@ubuntu:~$
```

FIGURE 3.10 : Lancement du conteneur Wordpress

Un extrait de la plateforme WordPress :

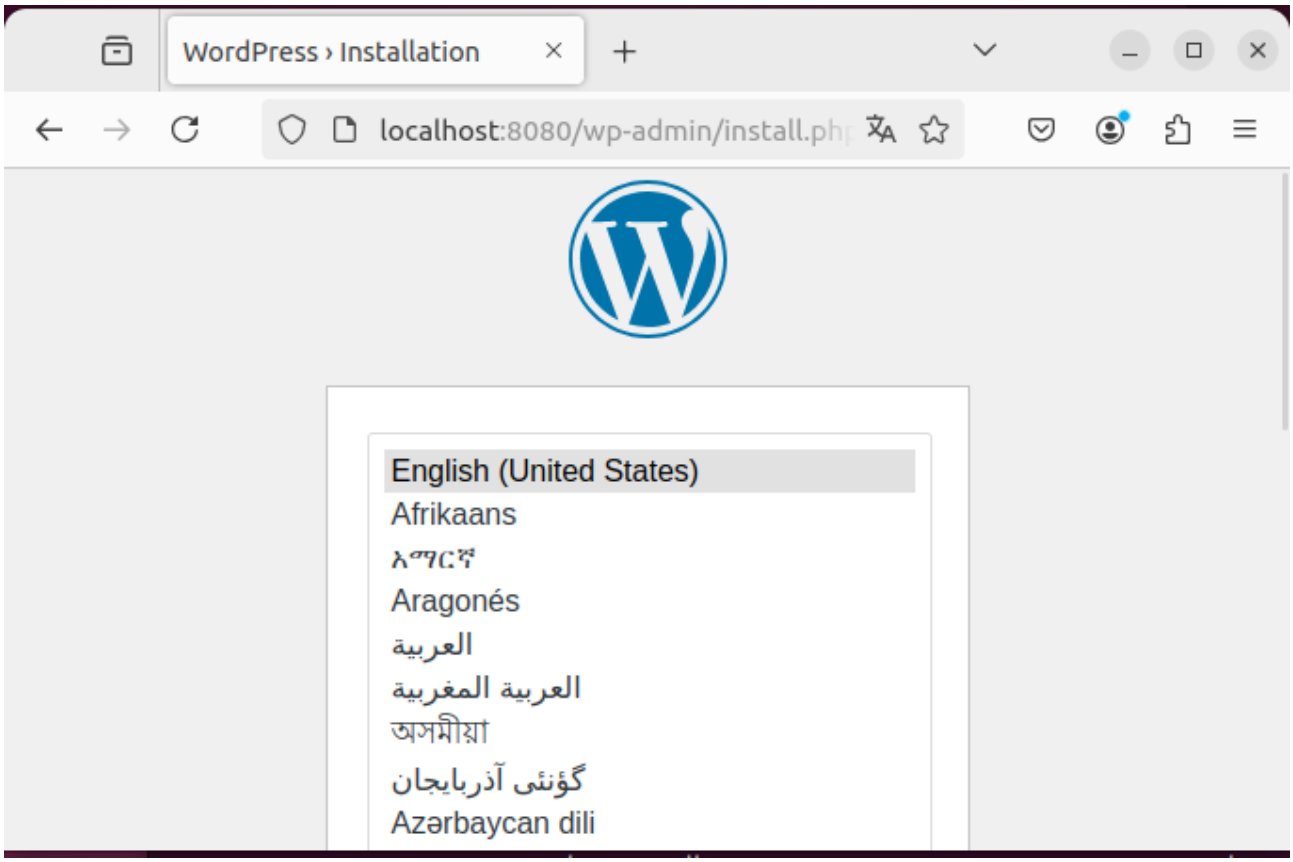


FIGURE 3.11 : Installation Wordpress

La page de connexion déployé sur la machine server_VPN :

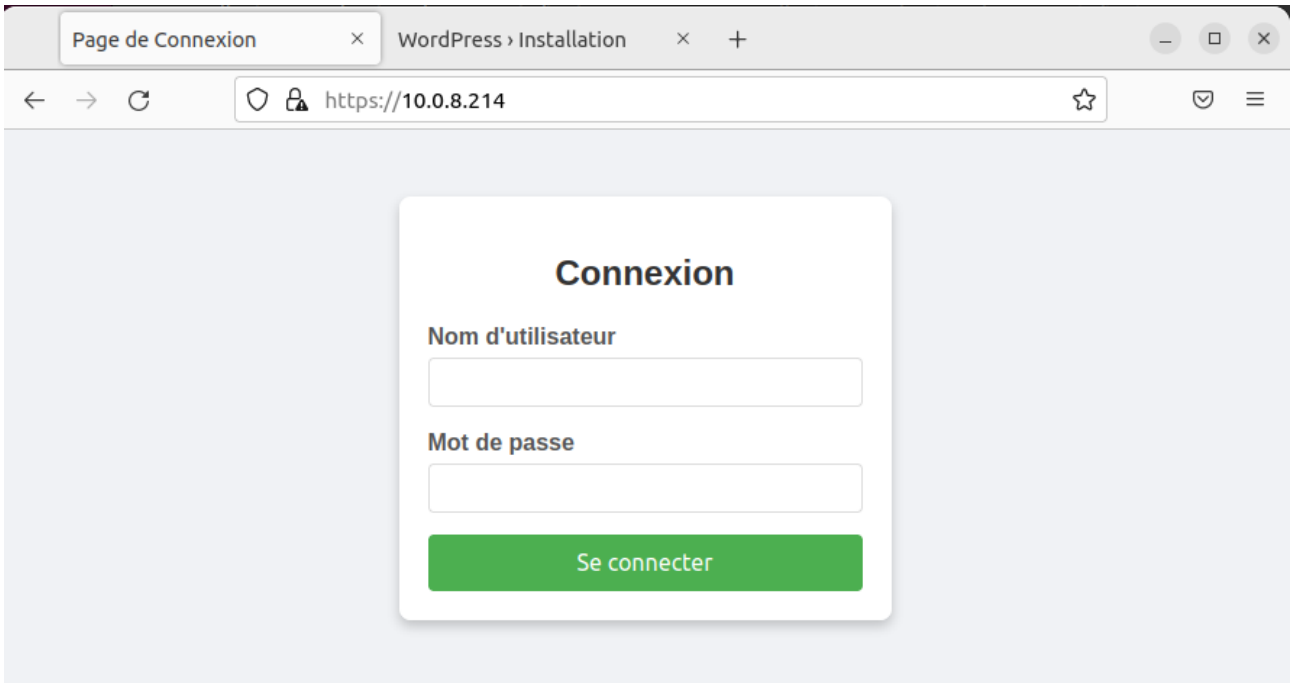


FIGURE 3.12 : Page de connexion

Pour démarrer la redirection de port dynamique, il convient de lancer le script **Automator_port_forwarding.s**
Ensuite, les informations suivantes doivent être fournies :

- L'adresse IP de la machine intermédiaire

- Le nom d'utilisateur de la machine intermédiaire
- Le port de sortie des requêtes de la machine cliente (dans notre cas, nous avons choisi l'option **port aléatoire**)

```
client_vpn@ubuntu:~/Documents/test$ ./Automator_port_forwarding.sh
Bienvenue dans le processus d'automatisation de la redirection de port.
Voulez-vous configurer une paire de clés SSH ? (o/n) n
Aucune configuration de clé SSH n'est nécessaire, poursuite du processus.
Entrez le chemin de la clé privée à utiliser (ou appuyez sur Entrée pour utiliser ) :
/home/client_vpn/Documents/test/client
Choisissez le type de port forwarding :
1) Local Port Forwarding
2) Dynamic Port Forwarding
Entrez le numéro correspondant : 2
Veuillez entrer l'IP de la machine intermédiaire : 192.168.122.1
Veuillez entrer le nom d'utilisateur de la machine intermédiaire : kali
Vous avez choisi la redirection de port dynamique.
Choisissez comment définir le port dynamique :
1) Port dynamique personnalisé
2) Port aléatoire (entre 9000 et 65535)
Entrez le numéro correspondant : 2
Un port dynamique aléatoire a été sélectionné : 29765
Lancement de la redirection de port dynamique : ssh -i /home/client_vpn/Documents/test
/client -D 29765 kali@192.168.122.1
Connexion au serveur réussie
```

FIGURE 3.13 : Execution du script Automator_port_fowrading dans le cas de la redirection dynamique

La configuration de la redirection étant terminée, pour accéder à la page de connexion ainsi qu'à la page de configuration de WordPress, il faut se rendre dans les paramètres du navigateur, activer l'option "**Configuration manuelle du proxy**" et renseigner le champ Hôte SOCKS avec "**localhost**" et le port "**29765**" (port choisi par le programme), puis valider les modifications.



FIGURE 3.14 : Paramétrage pour l'utilisation du proxy SOCKS au niveau du navigateur

Après avoir configuré le proxy SOCKS, l'accès à la page de connexion se fait en saisissant l'URL suivante dans la barre de recherche : `https://10.0.8.214`

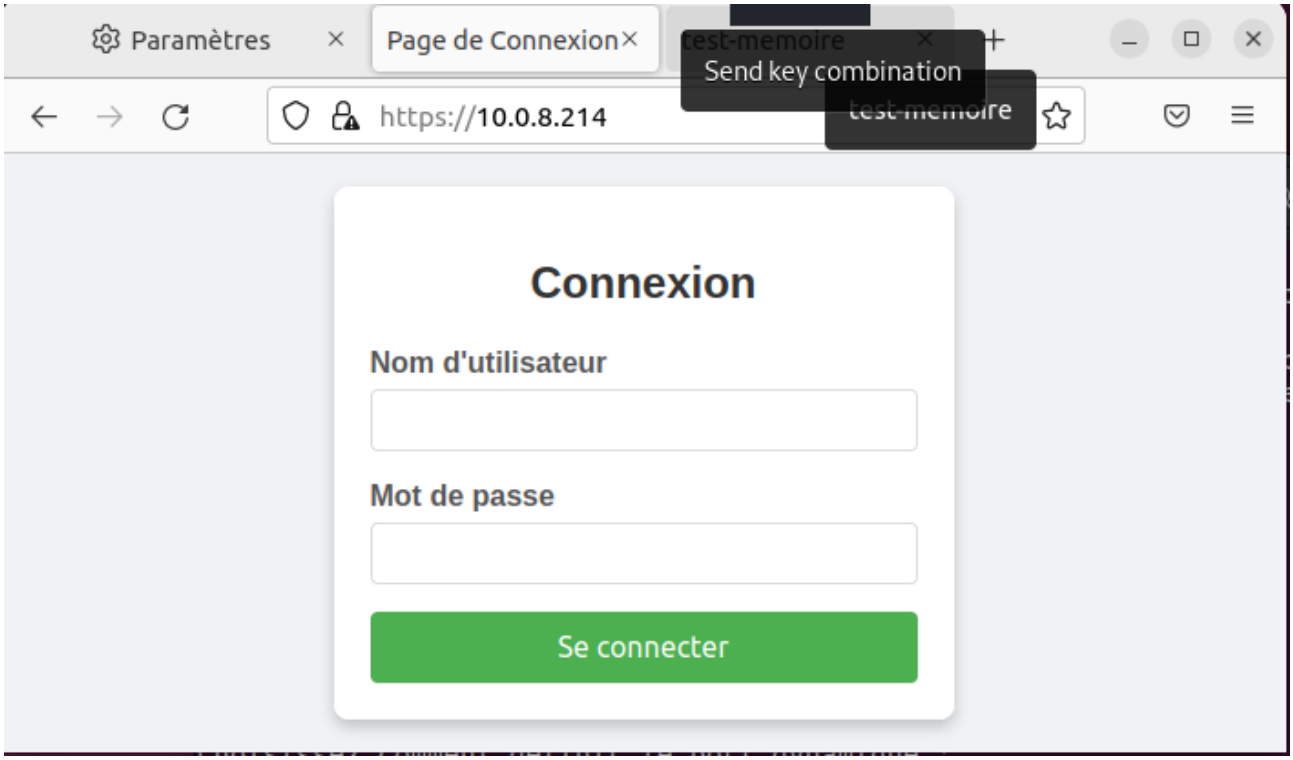


FIGURE 3.15 : Accès à la page de connexion depuis la machine cliente (redirection de port dynamique)

Concernant l'application WordPress, elle est accessible via le port 8080 de la machine virtuelle serveur. Pour y accéder, il suffit de saisir l'URL suivante dans la barre de recherche :

`http://10.0.8.214:8080/wp-admin/install.php`

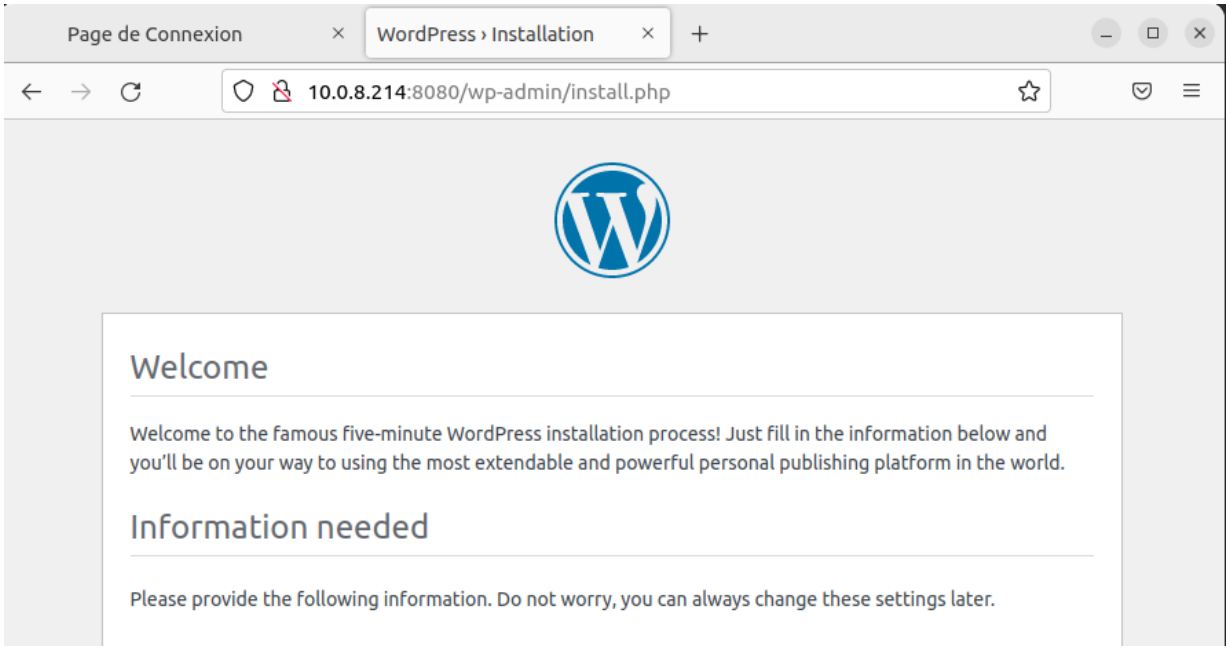


FIGURE 3.16 : Accès à l'application Wordpress depuis la machine cliente (redirection de port dynamique)

Cette approche permet au développeur, même depuis un emplacement distant, de poursuivre son travail sans compromettre la sécurité du réseau.

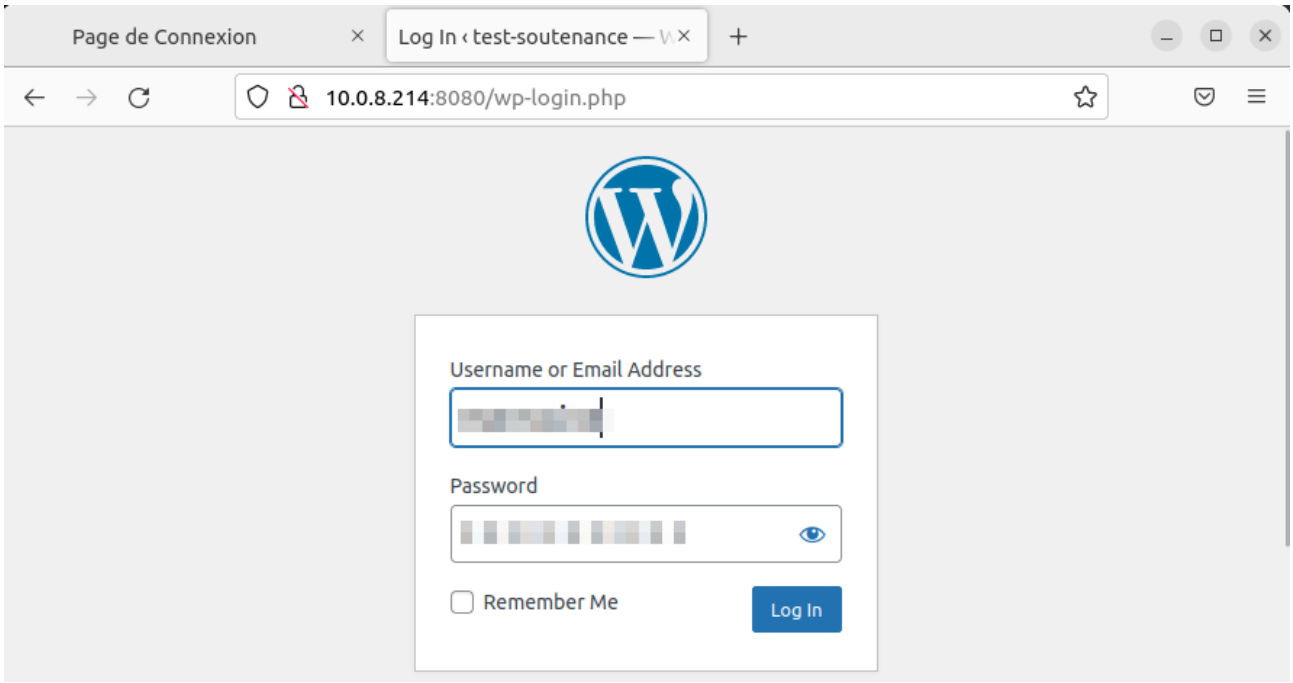


FIGURE 3.17 : Configuration Wordpress

On peut observer que la communication sur le réseau public est chiffrée grâce au protocole SSH.

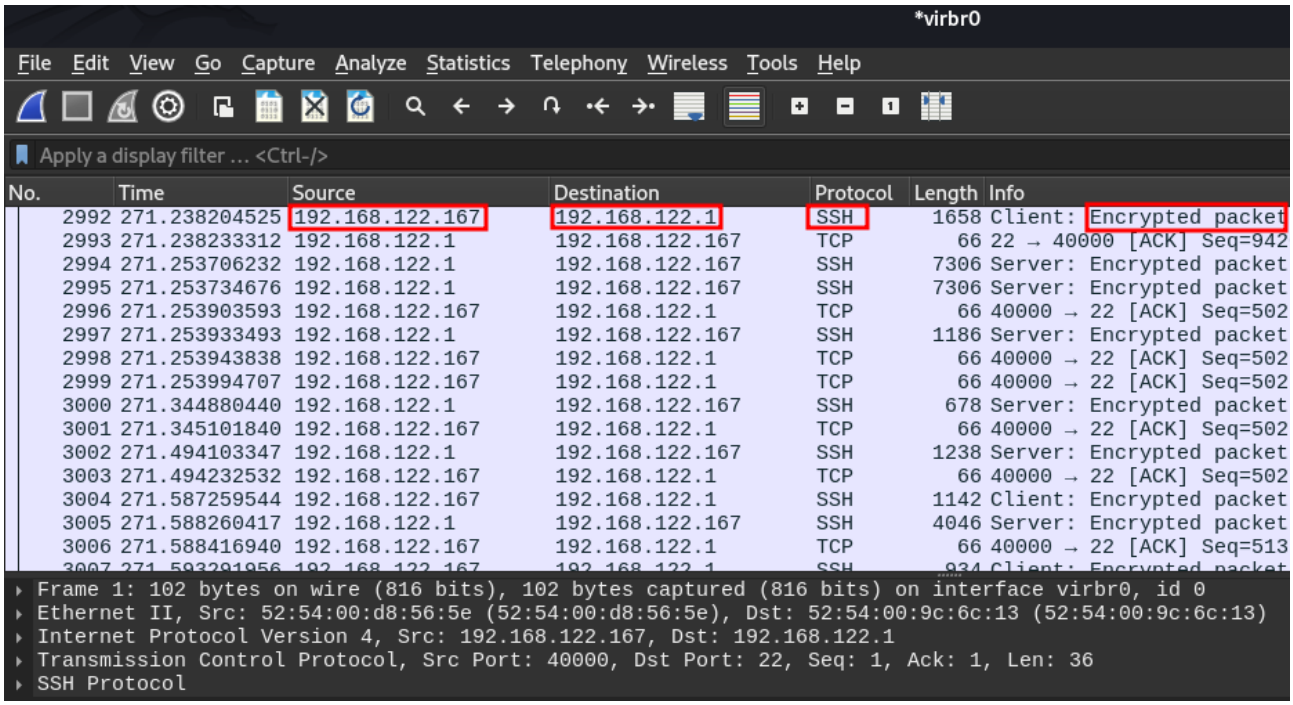


FIGURE 3.18 : Capture réseau avec Wireshark

Conclusion

En conclusion, ce chapitre a d'abord examiné le choix de l'outil de scripting, puis a détaillé le processus de création du script d'automatisation, qui propose trois types de redirection de ports. Enfin, nous avons présenté le fonctionnement du script, illustré par l'option de redirection de port local et de dynamique. Bien que ces démonstrations ne couvrent pas toutes les méthodes, le script est conçu pour offrir flexibilité et efficacité dans la gestion des connexions sécurisées via SSH, permettant ainsi un accès simplifié et sécurisé aux ressources réseau.

Conclusion Générale

Ce mémoire a exploré l'exploitation du port forwarding comme une application spécifique des techniques de tunneling, visant à optimiser les communications réseau. L'étude a mis en lumière comment la redirection de port, en tant que mécanisme de tunneling, permet d'améliorer l'efficacité et la flexibilité des réseaux en permettant l'accès à des services spécifiques à travers des tunnels sécurisés. L'analyse des différents types de tunneling et de leurs applications pratiques a montré que le port forwarding est un outil puissant pour surmonter les limitations des architectures réseau classiques.
En intégrant un script automatisé en BASH pour faciliter la configuration de la redirection de port, ce travail a optimisé l'aspect pratique de l'exploitation de ces techniques tout en assurant un haut niveau de sécurité grâce à des configurations SSH adaptées. L'étude a démontré que le port forwarding offre une solution simple, mais efficace, permettant aux entreprises et aux utilisateurs avancés d'exploiter pleinement les capacités des réseaux sans compromettre la sécurité.
Enfin, cette recherche ouvre la voie à de futures explorations sur l'optimisation des techniques de tunneling, notamment dans des contextes où les exigences de performance et de sécurité sont particulièrement élevées, tout en répondant aux défis d'un environnement numérique de plus en plus complexe.

Perspectives

À l'avenir, plusieurs pistes d'amélioration pourraient être envisagées pour enrichir et perfectionner l'application. Tout d'abord, le développement d'une interface graphique simplifierait considérablement l'utilisation de l'outil, en le rendant accessible à des utilisateurs non techniques et en facilitant la configuration des paramètres de redirection de port. Ensuite, une compatibilité multiplateforme serait un atout majeur, permettant à l'application de fonctionner de manière fluide sur différents systèmes d'exploitation tels que Windows, macOS et Linux. Par ailleurs, une optimisation de la gestion des utilisateurs pourrait être intégrée, offrant un contrôle précis des permissions et des accès, ce qui renforcerait la sécurité dans des environnements multi-utilisateurs. Enfin, des améliorations spécifiques à la prise en charge de scénarios complexes de redirection de port pourraient être apportées pour rendre l'application encore plus robuste et polyvalente. Ces évolutions contribueraient non seulement à élargir le public cible de l'application, mais aussi à répondre à des besoins variés tout en garantissant une meilleure efficacité, sécurité et adaptabilité de la solution proposée.

Bibliographie

[1] William Stallings. *Cryptography and Network Security : Principles and Practice*.7th ed., Pearson, 2017.

[2] R. Lemos. *Practical Linux Security Cookbook : Practical recipes to secure your Linux environment and keep your applications, data, and networks safe*. Packt Publishing, 2018.

Webographie

[3] The Hacker Recipes. Port forwarding. https://www.thehacker.recipes/infra/pivoting/port-forwarding. Consulté en Août 2024.

[4] Cybernews. What is port forwarding. https://cybernews.com/what-is-vpn/port-forwarding/. Consulté en Août 2024.

[5] VPN : avantages et inconvénients. https://www.journaldugeek.com/vpn/faq/avantages-inconvenients/. Consulté en Août 2024.

[6] Datawookie. SSH Tunnel : Dynamic Port Forwarding. https://datawookie.dev/blog/2023/12/ssh-tunnel-dynamic-port-forwarding/. Consulté Août 2024.

[7] https://itrus.su/itrus.su--linux_t&t--ssh--ssh_tunneling_and_port_forwarding_in_pictures_305.html. Consulté Août 2024.

[8] Kaspersky. What is a VPN. https://www.kaspersky.fr/resource-center/definitions/what-is-a-vpn. Consulté Septembre 2024.

[9] Github : mohammedAcheddad/SSH_port_forwarding. https://github.com/mohammedAcheddad/SSH_port_forwarding?tab=readme-ov-file#comparison-of-types. Consulté en Septembre 2024.

[10] FrameIP : Introduction aux VPN. https://www.frameip.com/vpn/#1-8211-introduction-au-reseau-prive-virtuel-vpn. Consulté en Octobre 2024.

[11] https://fr.m.wikipedia.org/wiki/Fichier:Internet_port_forwarding.png. Consulté en Octobre 2024

[12] AVG : Qu'est ce qu'un VPN ? https://www.avg.com/fr/signal/what-is-a-vpn-and-why-should-you-use-one.Consulté en Octobre 2024.

Annexe

Script complet Automator_port_forwarding.sh

Disponible sur le github : https://github.com/OthnielAD/Memoire.git

```bash
#!/bin/bash

# Message de bienvenue
echo "Bienvenue dans le processus d'automatisation de la redirection de port."
# Demande à l'utilisateur s'il souhaite configurer une paire de clés SSH
read -p "Voulez-vous configurer une paire de clés SSH ? (o/n) " CONFIGURE_KEYS
# Vérification et création des clés SSH si nécessaire
if [[ "$CONFIGURE_KEYS" =~ ^[oO]$ ]]; then
    KEY_PATH="$HOME/.ssh/id_rsa"

    echo "Les clés SSH seront enregistrées dans $KEY_PATH."
    # Création de la paire de clés
    read -p "Voulez-vous ajouter une passphrase à votre clé SSH ? (o/n) " RESPONSE
    if [[ "$RESPONSE" =~ ^[oO]$ ]]; then
        ssh-keygen -t rsa -b 2048 -f "$KEY_PATH"
    else
        ssh-keygen -t rsa -b 2048 -f "$KEY_PATH" -N ""
    fi
        echo "Clé SSH générée avec succès."
    echo "N'oubliez pas d'ajouter la clé publique à 'authorized_keys' sur la machine intermédiaire."
else
    echo "Aucune configuration de clé SSH n'est nécessaire, poursuite du processus."
fi

# Demande à l'utilisateur de choisir une clé
read -p "Entrez le chemin de la clé privée à utiliser (ou appuyez sur Entrée pour utiliser $KEY_PATH) : " USER_KEY_PATH
USER_KEY_PATH=${USER_KEY_PATH:-$KEY_PATH}  # Utilise la clé par défaut si aucun chemin n'est fourni
```

FIGURE 3.19 : Première partie du script (configuration des clés SSH et le choix de la redirection)

```bash
# Demande le type de redirection de port
echo "Choisissez le type de port forwarding :"
echo "1) Local Port Forwarding"
echo "2) Dynamic Port Forwarding"
read -p "Entrez le numéro correspondant : " PORT_FORWARDING_CHOICE

# Demande l'IP publique de la machine intermédiaire
read -p "Veuillez entrer l'IP de la machine intermédiaire : " INTERMEDIATE_IP

# Demande le nom d'utilisateur de la machine intermédiaire
read -p "Veuillez entrer le nom d'utilisateur de la machine intermédiaire : " USERNAME
```

FIGURE 3.20 : Deuxième partie du script (choix de la redirection de port)

```
# Gestion des choix de port forwarding
case $PORT_FORWARDING_CHOICE in
    1)
        # Redirection de port local
        echo "Vous avez choisi la redirection de port local."
        echo "Choisissez comment définir le port local :"
        echo "1) Port local personnalisé"
        echo "2) Port aléatoire (entre 9000 et 65535)"
        read -p "Entrez le numéro correspondant : " PORT_CHOICE
        if [ "$PORT_CHOICE" == "1" ]; then
            read -p "Veuillez entrer le port local : " LOCAL_PORT
            # Vérification de la validité du port
            if [[ $LOCAL_PORT -lt 9000 || $LOCAL_PORT -gt 65535 ]]; then
                echo "Port local non valide. Veuillez entrer un port entre 9000 et 65535."
                exit 1
            fi
        elif [ "$PORT_CHOICE" == "2" ]; then
            LOCAL_PORT=$((RANDOM % 56535 + 9000))  # Port aléatoire entre 9000 et 65535
            echo "Un port local aléatoire a été sélectionné : $LOCAL_PORT"
        else
            echo "Choix invalide. Veuillez relancer le script."
            exit 1
        fi
        # Demande l'IP de la machine cible du réseau intranet
        read -p "Veuillez entrer l'IP de la machine cible du réseau intranet : " TARGET_IP
        # Demande le port distant sur lequel tourne le service sur la machine cible
        read -p "Veuillez entrer le port distant sur lequel le service tourne sur la machine cible : " TARGET_PORT
        echo "Lancement de la redirection de port local : ssh -i $USER_KEY_PATH -L $LOCAL_PORT:$TARGET_IP:$TARGET_PORT $USERNAME@$INTERMEDIATE_IP"
        ssh -i "$USER_KEY_PATH" -L "$LOCAL_PORT:$TARGET_IP:$TARGET_PORT" "$USERNAME@$INTERMEDIATE_IP"
        ;;
```

FIGURE 3.21 : Troisième partie du script (configuration des paramètres de la redirection de port local)

```
2)
    # Redirection de port dynamique
    echo "Vous avez choisi la redirection de port dynamique."

    # Redirection de port distant
    echo "Vous avez choisi la redirection de port dynamique."
    echo "Choisissez comment définir le port dynamique :"
    echo "1) Port dynamique personnalisé"
    echo "2) Port aléatoire (entre 9000 et 65535)"
    read -p "Entrez le numéro correspondant : " PORT_CHOICE

    if [ "$PORT_CHOICE" == "1" ]; then
        read -p "Veuillez entrer le port dynamique : " REMOTE_PORT
        # Vérification de la validité du port
        if [[ $REMOTE_PORT -lt 9000 || $REMOTE_PORT -gt 65535 ]]; then
            echo "Port dynamique non valide. Veuillez entrer un port valide."
            exit 1
        fi
    elif [ "$PORT_CHOICE" == "2" ]; then
        REMOTE_PORT=$((RANDOM % 56535 + 9000))  # Port aléatoire entre 9000 et 65535
        echo "Un port dynamique aléatoire a été sélectionné : $REMOTE_PORT"
    else
        echo "Choix invalide. Veuillez relancer le script."
        exit 1
    fi

    echo "Lancement de la redirection de port dynamique : ssh -i $USER_KEY_PATH -D $REMOTE_PORT $USERNAME@$INTERMEDIATE_IP"
    ssh -i "$USER_KEY_PATH" -D "$REMOTE_PORT" "$USERNAME@$INTERMEDIATE_IP"
    ;;
```

FIGURE 3.22 : Quatrième partie du script (configuration des paramètres de la redirection de port dynamique)

```
*)
    echo "Choix invalide. Veuillez relancer le script."
    exit 1
    ;;
```

FIGURE 3.23 : Cinquième partie du script (gestion des entrées utilisateur)

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des tableaux viii

Liste des acronymes ix

Glossaire 1

Introduction 2

1 Revue de littérature 4

Introduction 4

1.1 Introduction au tunneling 4

1.1.1 Définition et principes fondamentaux 4

1.1.2 Importance du tunneling dans les réseaux modernes 5

1.1.3 1.1.3 Applications courantes du tunneling (VPN, SSH, etc.) 6

1.2 Présentation du port forwarding (redirection de port 7

1.2.1 Définition et concepts de base 7

1.2.2 Les types de port forwarding 8

1.2.3 Différences et similitudes avec d'autres techniques de tunneling 10

1.3 Avantages et limites du port forwarding 11

1.3.1 Sécurisation des communications réseau 11

1.3.2 Optimisation des performances réseau 11

1.3.3 Limitations et défis associés 12

Conclusion 12

2 Automatisation du port forwarding 14

Introduction 14

2.1 Présentation de la redirection de port 14

2.1.1 Fonctionnement du port forwarding dans le tunneling 14

2.2 Avantages du port forwarding pour les activités des petites entreprises et startups . . 15
2.3 Implémentation du script pour la redirection de port 16
2.3.1 Fonctionnalité du script 16
2.3.2 Outils choisis pour l'implémentation 16
2.3.3 Configuration SSH adoptée pour sécuriser les connexions 16
2.4 Mise en place du script de redirection 17
2.4.1 Redirection de port local 18
2.4.2 Redirection de port dynamique 20
2.4.3 Redirection de port distant 21
Conclusion 22

3 Test et fonctionnement du script de redirection de port 23
Introduction 23
3.1 Test et fonctionnement du script 23
3.1.1 Exemple d'utilisation de la redirection de port local 25
3.1.2 Exemple d'utilisation de la redirection de port dynamique 27
Conclusion 31

Conclusion 33

Perspectives 34

Bibliographie 35

Webographie 36

Annexe 37

Table des matières 39