REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université 20 Août 1955 Skikda

Faculté des sciences-Département de l'informatique

Mémoire de fin d'études en vue l'obtention du diplôme Master : Réseaux et Systèmes Distribués (RSD)

Thème :

Etude et mise en œuvre d'un honeypot

Réalisé par :

- Guembour Omar Anis
- Boulhout Djamel Eddine

Encadré par :

- Touil Ghassen

Année Universitaire : 2021 / 2022

Remerciement

Nous remercions avant tout, Dieu de nous avoir donné la force morale et physique et nous a permis de terminer ce travail.

Nous adressons nos remerciements les plus sincères, à monsieur le professeur, notre encadreur Touil Ghassen qui a guidé nos réflexions et a accepté de répondre à nos questions, pour sa patience, sa disponibilité et ses précieux conseils et sa critique constructive.

À nos familles pour leur soutien et leur encouragement.

Nous remercions aussi les membres du jury qui ont accepté d'y participer et de juger ce travail.

Merci.

Résumé

Aujourd'hui, le principal défi dans le domaine de la sécurité des réseaux est de suivre l'évolution des menaces comme elles évoluent quotidiennement pour assurer la meilleure solution de protection de n'importe quel système. Il existe de nombreux mécanismes de protection traditionnels comme les pare-feu et les systèmes de détection de pénétration, mais ils ne sont pas en mesure de détecter de nouveaux types d'attaques. Par conséquent, des pièges de piratage ont été crée pour servir à détecter des attaques inconnues.

Les honeypots sont des systèmes construits et configurés pour permettre la pénétration afin de faciliter le déploiement des ressources de l'attaquant, utiliser son temps et le distraire de véritables systèmes. Ils fournissent également un milieu de travail pour étudier les techniques et les méthodes utilisées par les intrus sur le système. Il est important de garder un œil sur l'ensemble des activités dans ce domaine car par définition, toutes les activités de cette zone sont des attaques.

Ce mémoire présente la technologie des pièges de pénétration pour simuler pleinement les services SSH et interagir avec le pirate afin d'aider à consommer les ressources de l'attaquant, utiliser son temps, extraire autant d'informations précieuses que possible sur l'attaquant et ses techniques et les outils logiciels utilisés et le distraire des systèmes actuels.On vas utiliser Honey Modern Network, un serveur central pour déployer et gérer les pièges des disjoncteurs réseau et le programme Box Virtual en utilisant un ensemble de machines virtuelles pour simuler des attaques sur des serveurs SSH (Honeypots). Le cadre de travail proposé a été réalisé par l'attaque de devinettes Force-Brute sur un protocole SSH en utilisant l'outil Nmap dans le système Linux-Kali et la connectivité directe avec les autres services.

Abstract

Today, the main challenge in the field of network security is to follow the evolution of threats as they evolve daily to ensure the best solution of protection of any system. There are many traditional protection mechanisms such as firewalls and penetration detection systems, but they are not able to detect new types of attacks. As a result, penetration traps have been created to detect unknown attacks.

Honeypots are systems built and configured to allow penetration, to facilitate deployment of the attacker's resources, use the attacker's time and distract them from real systems. They also provide a working environment to study the techniques and methods used by intruders on the system. It is important to keep an eye on all activities in this area since by definition; all activities in this area are attacks.

This brief introduces penetration trap technology to fully simulate SSH services and interact with the hacker to help consume the attacker's resources, use his time, extract as much valuable information as possible about the attacker and his techniques and software tools used and distract him from current systems. We're going to use Honey Modern Network, a central server to deploy and manage network breaker traps and the Box Virtual program by using a set of virtual machines to simulate attacks on SSH (Honeypots) servers. The proposed framework was achieved by attacking Force-Brute riddles on a SSH protocol using the Nmap tool in the Linux-Kali system and direct connectivity with other services.

ملخص

يتمثل التحدي الرئيسي اليوم في مجال أمن الشبكات في متابعة تطور التهديدات مع تطورها يوميًا وذلك لضمان أفضل حل لحماية أي نظام.هناك العديد من الضمانات التقليدية مثل جدران الحماية وأنظمة الكشف عن الاختراق، لكنها غير قادرة على اكتشاف أنواع جديدة من الهجمات. ونتيجة لذلك، تم إنشاء مصائد الاختراق للكشف عن الهجمات غير المعروفة.

honeypot هي أنظمة مبنية ومهيأة للسماح بالاختراق لتسهيل نشر موارد المهاجم، واستخدام وقت المهاجم وصرفها عن الأنظمة الحقيقية. كما أنها توفر بيئة عمل لدراسة التقنيات والطرق التي يستخدمها المتسللون على النظام. ومن المهم مراقبة جميع الأنشطة في هذا المجال لأن جميع الأنشطة في هذا المجال هي، بحكم تعريفها، هجمات.

يقدم هذا الموجز تقنية honeypot لمحاكاة خدمات SSH بالكامل والتفاعل مع المتسلل للمساعدة في استهلاك موارد المهاجم، واستخدام وقته، واستخراج أكبر قدر ممكن من المعلومات القيمة حول المهاجم وتقنياته وأدوات البرامج المستخدمة وصرفه عن الأنظمة الحالية. سنستخدمHoney Modern Network ، وهو خادم مركزي لنشر وإدارة مصائد كسر الشبكة وبرنامج Box Virtual باستخدام مجموعة من الآلات الافتراضية لمحاكاة الهجمات على خوادم. (SSH (Honeypotsتم تحقيق الإطار المقترح من خلال مهاجمة ألغاز Force-Brute على بروتوكول SSH باستخدام أداة Nmap في نظام Linux-Kali والاتصال المباشر مع الخدمات الأخرى.

Table des matières

Chapitre 2 : Les honeypots

Liste des figures :

Liste des tableaux

Liste des Abréviations

ACL: Access Control List.

ARP: Address Resolution Protocol.

CD-ROM: Compact Disk – Read Only Memory.

DDOS: Distributed Denial of Service.

DMZ: Demilitarized Zone.

DNS: Domain Name System.

DTK: Deception Toolkit.

FTP: File Transfer Protocol.

HTTP: Hyper Text Transfer Protocol.

ICMP: Internet Control Message Protocol

IDS: Intrusion Detection System.

IP: Internet Protocol.

IPS: Intrusion Prevention System.

MHN: Modern Honey network.

MITM: Man In The Middle.

NMAP: Network Mapper.

OS: Operating System.

SNMP: Simple Network Management Protocol.

SQL: Structured Query Language.

SSH: Secure Shell.

TCP: Transmission Control Protocol.

TFN: Tribal Flood Network.

UDP: User Datagram Protocol.

UML: User Mode Linux.

Introduction générale

De jour en jour, nous devenons liés aux outils informatiques et de plus en plus, la dépendance deviendra immense et avec le développement de l'utilisation d'internet, les entreprises ouvrent leur systèmes d'informatiques à leur partenaires ou fournisseurs, il est donc nécessaire de protéger et maitriser le contrôle d'accès et les droits des utilisateurs du système informatique contre les menaces.

Devant cette croissance accrue dans l'utilisation des réseaux et des systèmes informatiques et la nature sensibles des données manipulées pour les systèmes et pour pallier aux menaces et attaques, il est primordial de mettre en place des mécanismes de sécurité. Cette contre-mesure représente l'ensemble des actions mises en œuvre en prévention de la menace, pas uniquement des solutions techniques mais également de formation et de sensibilisation à l'intention des utilisateurs.

En dépit de l'importance majeure de ce sujet contemporain et comme les attaques représentent aujourd'hui le plus grand défi pour la plupart des entreprises et des établissements. Nous avons choisi le thème de notre thèse :" étude et mis en œuvre d'un honeypot ".

Contrairement aux différent outils de sécurité préventifs traditionnels, les honeypots adoptent une approche non-défensive de la sécurité, une approche qui consiste à prendre l'initiative d'attirer les attaquants en les invitant à explorer un système d'informatique à l'attaque et le corrompre tout en analysant dans le détail son comportement et c'est une nécessité de poser ces questions :

- C'est quoi la sécurité informatique et quels sont ses critères ?
- C'est quoi un honeypot ? Quelle est son importance et ses objectifs ?
- Quels sont les types du honeypot ?

Et pour réaliser ce mémoire, nous l'avons organisé comme suit :

- Le premier chapitre : s'intéresse à la sécurité informatique, les différentes attaques existantes à l'heure actuelle contre les systèmes d'information, et les solutions utilisées pour la détection d'attaques.
- Le deuxième chapitre : parle d'une façon générale sur les honeypots, leurs objectifs, classification, types et son architecture.
- Le troisième chapitre : déploiement du honeypot-cowrie dans notre réseau.
- Le dernier chapitre : simulation d'une attaque sur le Honeypot-cowrie.

1. Introduction :

De nos jours, l'outil informatique connaît des avancées très significatives et aussi des nombreuses menaces. Il existe deux grands types des menaces : les risques humains et les risques matériels. Donc il est nécessaire d'élaborer une politique de sécurité qui assure la protection des systèmes d'information contre les incohérences, les interruptions et les intrusions, pouvant atteindre ce système d'information.

Dans ce chapitre, nous allons présenter quelques notions reliées à la sécurité informatique et les différents types d'attaques.

2. La sécurité informatique :

La sécurité englobe toutes les ressources informatiques mises en œuvre pour réduire la vulnérabilité du système aux menaces accidentelles ou intentionnelles, permettant au système informatique de fonctionner normalement. Il s'agit également d'assurer les personnes qui modifient ou accèdent aux données du système sont autorisés et peuvent parce que le service est disponible. **[1]**

L'objectif de la sécurité informatique est de s'assurer que les ressources matérielles et/ou logicielles d'un parc informatique ne sont utilisées que dans le cadre prévu par du personnel habilité.

La sécurité informatique implique la protection des risques liés à l'informatique ; elle doit prendre en compte :

- Eléments à protéger : appareils, données, utilisateurs ;
- Leur vulnérabilité ;
- Leur sensibilité : charge de travail impliquée, confidentialité, etc.
- La menace qui pèse sur eux ;
- Traitement (prévention et traitement) : complexité de mise en œuvre, coût, etc… **[2]**

3. Les critères de sécurité :

Il convient d'identifier les exigences fondamentales en sécurité informatique, qui caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques au regard de la sécurité :

- **Disponibilité :** Cela implique d'assurer le fonctionnement normal du système et l'accès aux services et aux ressources en permanence. On mesure la disponibilité d'un appareil en divisant le temps au cours duquel l'appareil a fonctionné par le temps où il aurait dû fonctionner. **[3]**
- **Confidentialité :** Concerne la prévention de la divulgation non autorisée d'information. La divulgation pourrait être intentionnelle, comme la rupture d'un chiffre de données et lire l'information, ou cela pourrait être involontaire, en raison de la négligence ou de l'incompétence des personnes qui traitent les informations. **[4]**
- **Intégrité :** L'intégrité consiste à garantir trois buts principaux :
    - Préserver la modification des informations par les utilisateurs non autorisé.
    - Préserver la modification non autorisé ou involontaire d'information par les utilisateurs autorisés.
    - Préserver la cohérence interne et la cohérence externe:
        - La cohérence interne: consiste à garantir la cohérence des données interne. Par exemple dans une organisation on assure que le nombre total des articles maintenus par cette organisation est égal à la somme des mêmes articles dans la base de données.
        - La cohérence externe: consiste à assurer que la cohérence entre les données dans la base de données et le monde réel est maintenue. Par exemple dans une entreprise on assure que le nombre des articles vendus est le même nombre dans la base de données. **[4]**
- **Non-répudiation** : La non-répudiation comprend l'assurance qu'aucun correspondant ne peut nier la transaction. Ainsi, lorsqu'un message est envoyé, le destinataire peut prouver que le message a bien été envoyé par le prétendu expéditeur. De même, lorsqu'un message est reçu, l'expéditeur peut prouver que le message a bien été reçu par le destinataire prévu. **[5]**
- **Authentification** : L'authentification comprend la garantie de l'authenticité des communications. Dans le cas de messages élémentaires, tels que des signaux d'avertissement ou d'alarme, ou le déclenchement de commandes, la fonction d'un service d'authentification est d'assurer au destinataire que le message provient bien de la source dont il prétend provenir. **[6]**

4. Anatomie d'une attaque :

Cinq verbes anglophones appelés « les 5 P » dans la littérature constituent le squelette de toute attaque informatique **[7]**:

- **Probe** : Comprend la collecte d'informations via des outils tels que whois, Arin, les recherches DNS, etc. Les informations sur le système cible peuvent être recueillies de plusieurs manières, telles que l'analyse des ports avec le programme Nmap pour déterminer la version du logiciel utilisé, ou l'analyse des vulnérabilités avec le programme Nessus. Des outils tels que pare-feu, hping ou SNMP Walk vous permettent de découvrir la nature de votre réseau.
- **Penetrate** : Utilisez les informations collectées pour infiltrer le réseau. Des techniques telles que la force brute ou les attaques par dictionnaire peuvent être utilisées pour contourner la protection par mot de passe. Une autre possibilité d'infiltrer le système est d'exploiter les failles de l'application.
- **Persist** : Création d'un compte avec des droits super utilisateur afin qu'il puisse être réinfiltré ultérieurement. Une autre méthode consiste à installer une opération de contrôle à distance able à résister à un reboot (ex. : un cheval de Troie).
- **Propagate** : Cette étape consiste à observer l'accessibilité et la disponibilité du réseau local.
- **Paralyze** : Cette étape peut consister en plusieurs activités. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine, détruire des données ou reprises, endommager le système d'exploitation dans le but de grower le serveur.

5. Les malwares :

Un logiciel malveillant, également appelé logiciel indésirable ou programme malveillant ou spam ("malware"), est un programme développé pour nuire à un système informatique sans le consentement de l'utilisateur dont l'ordinateur est infecté. Aujourd'hui, le terme "virus" est souvent utilisé à tort pour désigner divers types de logiciels malveillants. Les malwares comprennent les virus, les vers, les chevaux de Troie et d'autres menaces. La catégorie de virus informatiques la plus populaire pendant longtemps a cédé la place aux chevaux de Troie en 2005. Les malicieux peuvent être classifiés en fonction des trois mécanismes suivants **[2]** :

- mécanisme de propagation (par exemple, un ver se propage sur un réseau informatique en exploitant une faille applicative ou vulnérabilité humaine) ;
- mécanisme de déclenchement (par exemple, la bombe logique comme la bombe logique baptisée vendredi 13 déclenche quand un événement se produit);
- charge utile (par exemple, le virus de Tchernobyl essaie de supprimer des parties importantes du BIOS, empêchant le démarrage de l'ordinateur infecté).

Les logiciels malveillants peuvent être classés suivant leur mode de corruption du noyau du système d'exploitation à quatre types **[8]**:

- **Type 0 Malware** :

  Il ne doit pas être considéré comme un malware du point de vue de la détection de compromission du système, il n'interagit avec aucune partie du système d'exploitation (ni autres processus) en utilisant des méthodes non documentées.

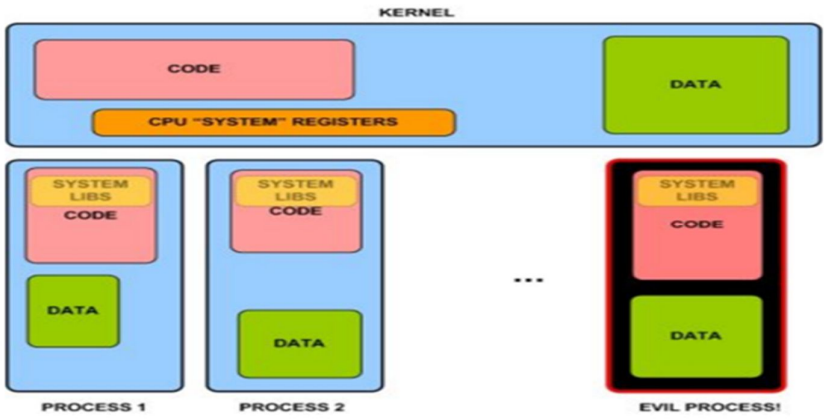

Figure 1.01: Type 0 malware.

- **Type I Malware** :

  Il modifie des choses qui ne devraient jamais être modifiées, comme code, ce qui en fait relativement facile à détecter.

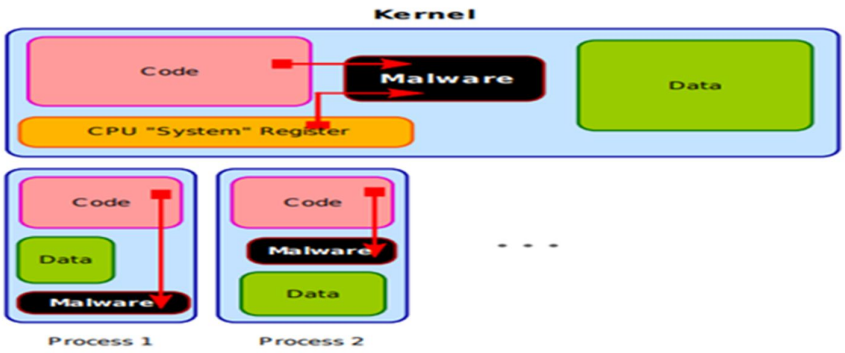

Figure 1.02: Type I malware.

- **Type II Malware :**

  Les logiciels malveillants de type II ne fonctionnent que sur des ressources dynamiques, comme les données section. En modifiant certains pointeurs de fonction dans certaines structures de données du noyau, de sorte que le code de l'attaquant est exécuté à la place du système d'origine ou du code d'application.

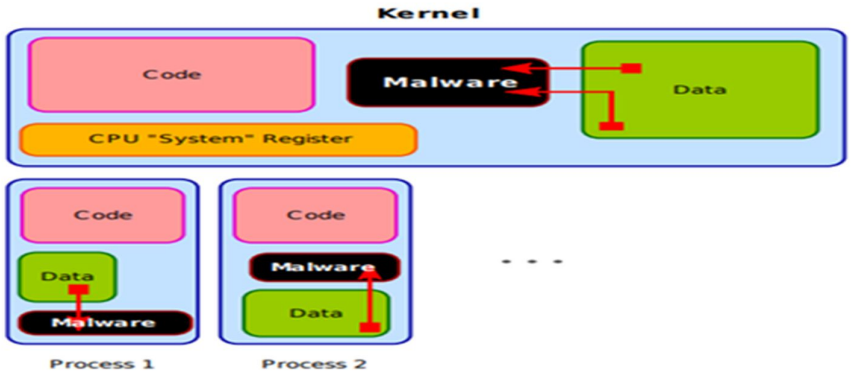

Figure 1.03: Type II Malware**.**

- **Type III malware** :

  A première vue, il semble similaire au malware de type 0, car il ne modifie en aucune façon la mémoire du système ni de l'application, mais en fait, il est très différent, car il permet de prendre le contrôle total du système en cours d'exécution et interfèrent avec lui. Les exemples actuels de logiciels malveillants de type III utilisent la technologie de virtualisation matérielle, mais nous pouvons imaginer qu'à l'avenir, une autre technologie sera introduite qui permettrait également la création d'un autre type de malware de type III.

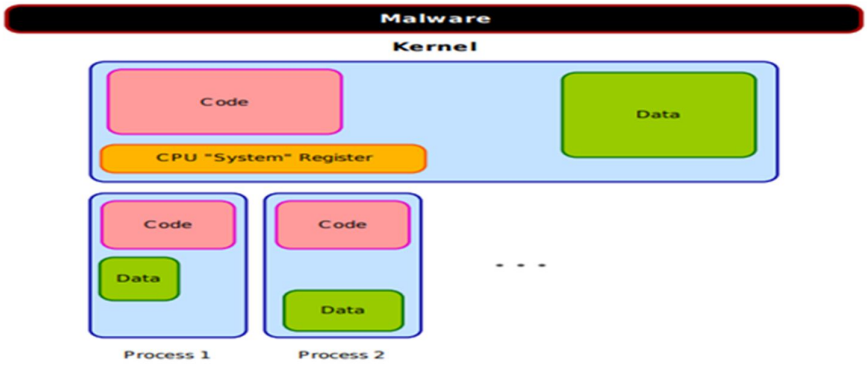

Figure 1.04: Type III malware.

Il existe des millions de malwares différents, on marque quelque types de malwares: virus, spam, vers réseau, porte dérobée, cheval de Troie, bombe logique.

5.1. Les virus :

Un virus est un programme qui s'exécute sur un ordinateur et qui peut se propager sur d'autres machines d'un système d'information dans le but de perturber les applications informatiques. Il existe plusieurs types de virus qui correspondent à des menaces plus ou moins graves. **[9]**

5.2. SPAM :

Il s'agit de messages électroniques non sollicités envoyés en grand nombre. Il se transmet à partir de l'utilisation de relais de messagerie ouverts ou de machines mal sécurisées. **[9]**

5.3. Vers réseau :

Un ver (Worm) est une variété de virus qui se propage par le réseau. Il peut s'agir d'un bot. En fait, alors qu'il y a cinq ou six ans les virus n'étaient pas des vers et les vers n'étaient pas des virus, aujourd'hui la confusion entre les deux catégories est presque totale. **[10]**

5.4. Porte dérobée :

Les portes dérobées sont des logiciels de communication cachés, tels que ceux installés par des virus ou des chevaux de Troie, qui permettent à des attaquants extérieurs d'accéder au réseau d'un ordinateur victime. **[10]**

5.5. Les chevaux de Troie :

Chevaux de Troie sont une forme de logiciels malveillants déguisés comme des logiciels utiles. Leur but : être exécuté par l'utilisateur, ce qui lui permet de contrôler l'ordinateur et de l'utiliser à des fins personnelles. En général, d'autres malicieux seront installés sur votre ordinateur, par exemple en autorisant la collecte, la falsification ou la destruction frauduleuse de données. **[6]**

5.6. Bombe logique :

Une bombe logique est un virus dans l'attente d'un évènement. Cet événement, déterminé par le programmeur malveillant, peut être une date particulière, une combinaison de touches, une action spécifique ou un ensemble de conditions précises. Un employé mal intentionné peut implanter une bombe logique chargée de vérifier si son nom disparaît des listes du personnel de l'entreprise ou son compte d'un serveur et nuire à l'entreprise après qu'il l'a quittée en détruisant ou corrompant des données. **[11]**

6. Les attaques :

Il existe plusieurs types d'attaques informatiques, elles se différencient les unes des autres selon le type d'actif visé, la méthode exploitée, et l'objectif fixé. **[12]**

6.1. Les attaques réseaux :

Les attaques réseaux s'appuient sur des failles liées directement aux protocoles ou à leur implémentation. Il existe 6 attaques réseaux plus connus:

6.1.1. **Fragments attacks** :

Cette attaque contourne la protection des dispositifs de filtrage IP. Pour sa mise en œuvre, le hacker utilise deux méthodes : Tiny Fragments et Fragment Overlapping. Parce que ces attaques sont historiques, les pare-feux actuels ont été implémentés avec une longue réflexion. **[13]**

6.1.2. **IP Spoofing** :

L'objectif de cette attaque est d'usurper l'adresse IP de la machine. Cela permet à un pirate de cacher la source de son attaque (pour déni de service) ou d'exploiter la relation de confiance entre les deux machines. Le principe de base de cette attaque est de forger ses propres paquets IP (à l'aide de programmes comme hping2 ou nemesis), où les pirates modifieront l'adresse IP source, etc. IP Spoofing est souvent appelé Blind Spoofing (ou Blind Spoofing). En effet, l'éventuelle réponse du paquet envoyé ne peut pas atteindre la machine du pirate car la source est falsifiée. Alors ils sont allés tromper la machine. **[13]**

Figure 1.05: Les étapes de l'IP Spoofing.

6.1.3. **TCP Session Hijacking :**

Le détournement de session TCP est utilisé pour rediriger les flux TCP. Les pirates peuvent alors contourner la protection par mot de passe (par exemple Telnet ou ftp). La nécessité d'une écoute passive (sniffing) limite la portée de cette attaque au réseau physique de la cible.

Voici les principaux points de compréhension de l'attaque. L'en-tête TCP se compose de plusieurs champs :

- Ports source et destination, identifiant la connexion entre les deux machines ;
- Un numéro de séquence identifiant chaque octet envoyé ;
- Le numéro d'acquittement correspondant au numéro d'acquittement du dernier octet reçu
- Les flags, comme :
    - SYN qui synchronise les numéros de séquence lors de l'établissement d'une connexion;
    - ACK, le flag d'acquittement d'un segment TCP;
    - PSH qui indique au récepteur de remonter les données à l'application.

La figure ci-dessous schématise l'établissement d'une connexion TCP (Three Way Handshake). **[13]**

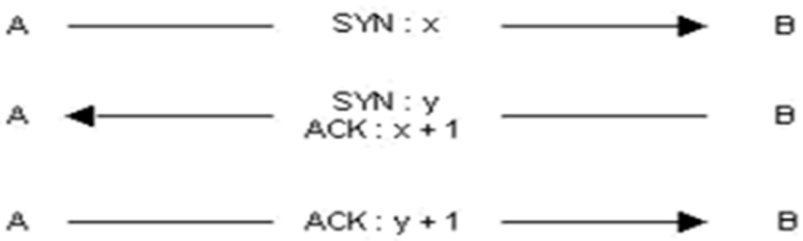

Figure 1.06: Protocole de contrôle de transmission.

6.1.4. ARP Spoofing:

Également connue sous le nom de redirection ARP, cette attaque redirige le trafic réseau d'une ou plusieurs machines vers la machine de l'attaquant. Cela se produit sur le réseau physique de la victime. Du fait de cette redirection, une personne malveillante peut se faire passer pour une autre personne. De plus, le pirate peut rediriger les paquets qu'il reçoit vers le vrai récepteur, de sorte que l'utilisateur trompé ne remarque rien. Même objectif que l'usurpation d'adresse IP, mais au niveau de la couche liaison de données. **[14]**

6.1.5. DNS Spoofing:

L'objectif de cette attaque est de rediriger les internautes vers des sites web à leur usurpation. Pour ce faire, les pirates exploitent une faiblesse du protocole DNS (domaine système de noms) et/ou son implémentation via des serveurs de noms. Pour rappel, le protocole DNS implémente la Correspondance entre l'adresse IP et le nom de la machine (par exemple :

www.truc.com). Il existe deux attaques majeures de DNS Spoofing : DNS ID Spoofing et DNS Cache empoisonnement. Plus précisément, l'objectif du pirate est de faire correspondre une adresse IP La machine qu'il contrôle est le nom réel et effectif de la machine publique. **[15]**

6.2. Les attaques applicatives :

Les attaques applicatives sont principalement basées sur des vulnérabilités spécifiques à l'application utilisée. Ces pannes peuvent être de différentes natures : problèmes de configuration, problèmes de code logiciel, problèmes liés à des erreurs d'interprétation de commandes ou d'exécution de scripts.

6.2.1. Les problèmes de configuration :

Afin de faciliter le fonctionnement du logiciel, il n'est généralement pas sûr d'utiliser la configuration par défaut (ex : le login/mot de passe par défaut pour le serveur de base de données). De plus, des erreurs peuvent se produire lors de la configuration du logiciel. Des serveurs mal configurés peuvent entraîner l'accès à des fichiers importants ou compromettre l'intégrité du système d'exploitation. C'est pourquoi il est important de lire attentivement la documentation fournie par les développeurs pour éviter les bogues. **[7]**

6.2.2. Les bugs :

Associés à des problèmes dans le code source, ils pourraient conduire à des exploits. Il n'est pas rare de voir une machine en action après une simple erreur de programmation. Cependant, vous ne pouvez pas faire grand-chose à propos de ces problèmes, à part attendre que les développeurs le résolvent. **[7]**

6.2.3. Un buffer overflow :

Un buffer overflow est une attaque très efficace et assez complexe à exécuter. Il vise à exploiter les vulnérabilités, les faiblesses des applications (type de navigateur, logiciel de messagerie, etc...) pour exécuter du code arbitraire qui compromettrait la cible (obtenir le droit administrateur, etc…). **[15]**

6.2.4. Les Script :

Malheureusement, une mauvaise programmation de scripts ou l'utilisation de fonctions boguées peut être source de failles de sécurité. Il convient d'être très attentif au niveau du développement d'un script. **[16]**

6.2.5. Man in the middle :

Attaquer un "man-in-the-middle" (littéralement "man-in-the-middle" ou "interceptor's attack"), parfois appelé MITM, est un scénario d'attaque dans lequel des pirates écoutent les communications entre deux interlocuteurs et forger des échanges pour se faire passer pour les parties. La plupart des attaques de l'homme du milieu impliquent d'écouter le réseau à l'aide d'un outil appelé renifleur. **[17]**

6.2.6. Injection SQL :

L'injection SQL est devenue un problème courant affectant les sites Web basés sur des bases de données. Cela se produit lorsque les criminels exécutent des requêtes SQL sur la base de données avec des données entrantes du client vers le serveur. Les commandes SQL sont insérées dans l'entrée du plan de données (par exemple, à la place du nom d'utilisateur ou du mot de passe) pour exécuter des commandes SQL prédéfinies. Une vulnérabilité d'injection SQL réussie peut lire des données de base de données sensibles, modifier (insérer, mettre à jour ou supprimer) des données de base de données, effectuer des opérations de gestion de base de données (comme la fermer), récupérer le contenu d'un fichier spécifique et, dans certains cas, envoyer une commande à système opérateur. **[18]**

6.3. Les dénis de service :

Le Déni de Service est une attaque pour rendre indisponible un service (application spécifique) ou la machine concernée. Nous distinguerons deux types de déni de services, d'une part ceux dont l'origine est l'exploitation d'un bug d'une application et d'autre part ceux dus à une mauvaise implémentation d'un protocole ou à des faiblesses de celui-ci.

6.3.1. Les dénis de service réseaux :

Il existe différents types de déni de service utilisant les spécificités des protocoles de la pile TCP/IP. **[13]**

6.3.2. UDP Flooding :

Ce refus de service utilise le mode hors connexion du protocole UDP. Il crée une "tempête de paquets UDP" (génère beaucoup de paquets UDP), soit pour une machine ou entre deux machines. Une telle attaque entre deux machines peut se traduire par une congestion du réseau et une surcharge des ressources sur les deux hôtes victimes. La congestion est plus grave parce que le trafic UDP a préséance sur le trafic TCP. En effet, le protocole TCP dispose d'un mécanisme de

contrôle de congestion qui réduit le trafic en s'adaptant à la fréquence de transmission des paquets TCP dans le cas où l'accusé de réception d'un paquet arrive après un long délai. Le protocole UDP n'a pas ce mécanisme, donc au bout d'un moment, le trafic UDP prend toute la largeur de bande, ne laissant qu'une petite partie au trafic TCP. **[13]**

6.3.3. SYN Attaque (ou TCP SYN flooding) :

Exploite les trois étapes de connexion du protocole TCP. Son rôle est de rendre indisponible un service TCP offert sur une machine en envoyant un grand nombre de requêtes au serveur ciblé. Le principe de cette attaque est de créer des connexions semi-ouvertes sur la machine cible afin de remplir la file d'attente où sont enregistrées les requêtes d'ouvertures de connexions. **[12]**

6.3.4. Smurfing :

Cette attaque fait appel au protocole ICMP. Quand un ping (message ICMP ECHO) est envoyé à une adresse de diffusion (par exemple 10.255.255.255), il est multiplié et envoyé à chaque machine dans le réseau. Le principe de l'attaque est de spoofer les paquets ICMP ECHO REQUEST envoyés en réglant l'adresse IP source de la cible**.** Le pirate envoie un flux continu de ping à l'adresse de diffusion du réseau, puis toutes les machines répondent à la cible avec des messages ÉCHO REPLY ICMP. Multipliez alors le trafic par le nombre d'hôtes à l'intérieur du réseau. Dans ce cas, l'ensemble du réseau cible subit un déni de service parce que le volume élevé de trafic généré par une telle attaque peut conduire à la congestion du réseau. **[13]**

6.3.5. Déni de service distribué :

Le déni de service se propage sur le réseau de la victime en attaquant à l'aide de plusieurs sources (démons) et en utilisant les maîtres qui les contrôlent. Les outils DDoS (Distributed Denial of Service) les plus connus sont Tribal Flood Network (TFN), TFN2K, Trino et Stacheldraht. La figure montre un réseau DDoS typique. **[13]**

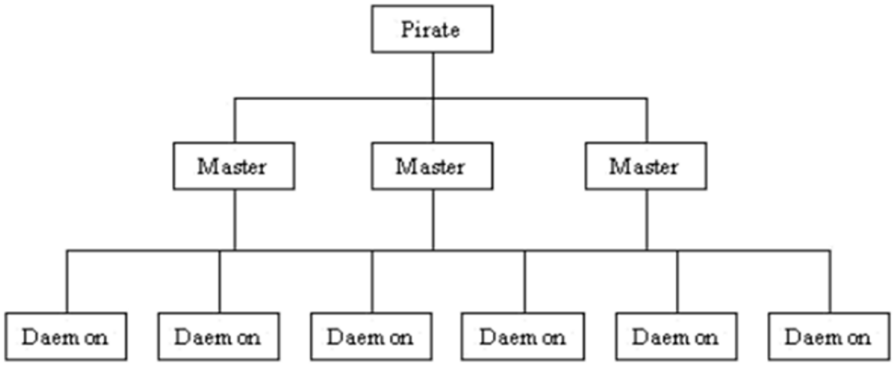

Figure 1.07: Un réseau typique de DDoS.

7. La détection d'attaque :

Afin de maintenir la protection du réseau informatique, il existe plusieurs méthodes pour éviter le risque des intrusions comme les anti-virus, les pare-feux, les systèmes de détection d'intrusions et la cryptographie.

7.1. Les anti-virus :

Un logiciel antivirus est un logiciel qui protège votre ordinateur contre les virus. Le logiciel antivirus est basé sur un fichier de signature et compare ensuite la signature génétique du virus au code à contrôler. Certains programmes appliquent également l'heuristique de découvrir les logiciels malveillants grâce à leur comportement. Les logiciels anti-virus peuvent analyser le contenu de votre disque dur, ainsi que la mémoire de votre ordinateur. Pour les plus modernes, ils interviennent en amont de la machine en contrôlant les échanges de fichiers (liaisons montantes et descendantes) avec l'extérieur. Par conséquent, vérifiez votre courriel et copiez à partir d'un support amovible comme un CD-ROM, une disquette, une connexion réseau. **[19]**

7.2. Pare-feux :

Un pare-feu (Firewall) est un système physique ou logique qui inspecte les paquets entrants et sortants du réseau afin d'autoriser ou d'interdire leur passage en se basant sur un ensemble de règles appelées ACL. On distingue trois principaux types de pare-feu :

- Pare-feu de filtrage de paquets : ce pare-feu filtre les paquets à l'aide de règles statiques qui testent les champs de protocole jusqu'au niveau de transport.

- Pare-feu de filtrage de paquets avec mémoire d'état : ce modèle garde les informations sur les services utilisés et les connexions ouvertes dans un tableau d'état. Il détecte ensuite des situations anormales suite à des infractions aux normes du protocole,
- Pare-feu proxy : ce pare-feu agit comme une passerelle d'application. En analysant les données jusqu'au niveau applicatif, il est en mesure de valider les demandes et les réponses pendant l'exécution des services réseau. **[20]**

7.3. La cryptographie :

La cryptographie est la transformation de données sous une forme qui ne peut pas être lue par quelqu'un qui ne dispose pas d'une méthode de déchiffrement (généralement une clé de déchiffrement). Son but est d'assurer la confidentialité en cachant des informations à toute personne qui ne devrait pas savoir. Cependant, la mise en place de moyens permettant aux informations de passer d'une personne à une autre n'élimine pas complètement les problèmes de sécurité. En effet, il faut aussi s'assurer de l'intégrité des opérateurs des serveurs utilisés pour la transmission sur Internet. De même, lorsqu'un client fournit son numéro de carte bancaire à un commerçant par exemple, il doit lui faire confiance. La cryptographie protège les données en assurant la transmission d'un ordinateur à un autre. Cette technologie ne protège en aucun cas les clients d'éventuelles fraudes de la part des commerçants ou des administrateurs de serveurs. **[21]**

7.4. Les systèmes de détection d'intrusion :

La détection d'intrusion est le processus de surveillance des événements qui se produisent dans un système ou un réseau informatique et de leur analyse pour détecter les signes d'intrusion, définis comme des signes de tentatives de compromettre la confidentialité, l'intégrité, la disponibilité ou le contournement des mécanismes de sécurité d'un ordinateur ou d'un réseau.. Les systèmes de détection d'intrusion sont des logiciels ou du matériel qui automatisent le processus de contrôle et d'analyse. Parmi ces systèmes les honeypots qui ont comme rôle majeur l'attraction des attaquants afin de les identifier, les faire tromper pour les empecher à accéder le système. Il existe différents emplacement, A cause de leur simplicité et leur cout, les honeypots sont utilisés à l'échelle mondiale dans sécurité informatique. **[22]**

8. Conclusion :

La sécurité informatique est primordiale. Elle sert à protéger les systèmes contre tout type de menace (malwares, les attaques réseaux, les attaques applicatives, les dénis de service). Il existe des

nombreuses méthodes de protection comme : Les anti-virus, les pare-feux, les systèmes de détection d'intrusions et la cryptographie.

Dans le prochain chapitre, nous allons présenter un système, relativement récent, introduite parmi les solutions de sécurité informatique « HoneyPot ».En commençant par la définition et les objectives. Ensuite, les différents types des honeypots, les classifiés selon leur intérêt d'utilisation et les interactions qu'ils permettent.

1. Introduction :

De nos jours, les attaques informatiques sont de plus en plus fréquentes. Donc, il est strictement important de développer des outils de protection afin de protéger les réseaux. Parmi les systèmes de protection développées « LE HONEYPOT » qui un système rendu volontairement vulnérable afin d'attirer les attaquants, observer leurs techniques et récupérer leurs outils.

2. Définition de honeypot :

Un honyepot est une ressource de sécurité dont la valeur réside dans la détection, l'attaque ou la compromission. **[23]**Un honyepot est un outil de sécurité qui doit être exploré et attaqué par un auteur de menaces. Il existe de nombreuses raisons pour lesquelles nous voudrions utiliser un tel outil. Une raison importante d'avoir un honeypot est d'étudier les modèles d'auteurs de menaces afin d'élaborer une variété de défenses contre les attaques. **[24]**

3. Objectifs :

Un honeypot est un système leurre qui peut être connecté directement à Internet ou être situé dans le réseau de l'entreprise. Son but est de permettre :

- D'étudier la nature du trafic (d'attaque) à destination de ce système.
- En cas de compromission, analyser les données générées par l'activité du hacker et prendre connaissance des outils, tactiques et motivations des hackers.
- Réduire de façon significative le taux de fausses alertes et le nombre d'attaques non détectées habituellement associées à l'utilisation de systèmes de détection des intrusions (IDS).
- Mesurer l'activité illégale ou anormale afin d'ajuster le niveau de protection requis pour la sécurité du réseau de l'entreprise. **[25]**

4. Types de honeypots :

Selon son but principal, les Honeypots peuvent être divisés en deux catégories distinctes : Honeypot de production et Honeypot de recherche.

4.1. Honeypot de production :

Les honeypots de production sont des pots à faible interaction avec peu d'interaction avec des assaillants ou des intrus dans le contexte. En outre, ils insistent moins sur la sécurité des moyens de production. Ils tentent de créer un environnement aussi irréel que possible. Quand ils sont déployés, ils n'ont pas nécessairement à simuler le système entier, mais autant que possible dans un certain laps de temps et de valeur. Une fois déployés, ils ne sont plus nécessaires. Ils saisissent des données. Essentiellement, ils ressemblent à un registre d'événements de base, avec une différence possible, et ils ne devraient pas interagir avec elle. Par exemple, pour surveiller les attaques Web, il suffit de simuler un serveur Web comme Apache et d'écouter les connexions du port 80. Une fois terminé, toutes les connexions qui scannent le honeypot pour les vulnérabilités HTTP seront sauvegardées. **[26]**

4.2. Honeypot de recherche :

L'objectif de ce type de honeypot n'est pas de protéger un système particulier, mais d'entrer dans un environnement de recherche pour comprendre et étudier comment la communauté black hat a évolué, quelles technologies sont utilisées par cette communauté et qui appartient à cette communauté. Les honeypots de recherche sont plus complets que les honeypots de production. Ce sont souvent des systèmes entiers (plutôt que des services individuels) qui peuvent être attaqués, ce qui en fait des systèmes sensibles à gérer et complique l'analyse de leurs résultats. L'étude des honeypots ne sert pas directement à la sécurité du système, mais elle fournit des informations précieuses sur les attaquants et leur comportement. Ces informations permettent de mieux comprendre la communauté des hackers, ce qui aide les professionnels de la sécurité informatique à améliorer les méthodes et les mécanismes de protection. **[27]**

5. Classification des honeypots :

5.1. Les honeypots à faible interaction :

Ils se caractérisent par une interaction minimale avec les pirates et l'imitation des faux services **[28]**. Ces types de honeypots généralement imitent des services spécifiques comme FTP ou HTTP. Il n'y a pas de système ou de service en cours d'exécution réel sur eux. Ce ne sont que des émulations exécutées au-dessus de la couche du système d'exploitation. Ils sont également plus simples à déployer et à entretenir, mais ne consignent que des informations limitées sur les activités de piratage. Puisqu'ils fonctionnent au-dessus de la couche du système d'exploitation, ils protègent le système des attaquants. Le maximum de dégâts qu'un attaquant peut faire est ce qu'il peut

encaisser la simulation de honeypot. Les honeypots à faible interaction peuvent être utilisés pour identifier les adresses IP des attaquants. Les honeypots à faible interaction les plus connus sont : Honeyd, Specter. **[29]**

Le fonctionnement général de ce type de Honeypot est illustré par la figure 2.01.

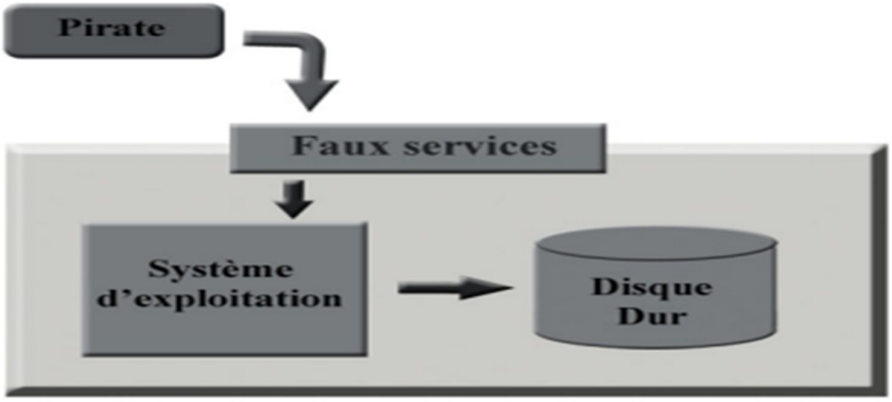

Figure 2.01: Schéma d'une interaction faible.

Les avantages et les inconvénients principaux de ce type de honeypots sont **[30]** :

5.1.1. Avantages :

- Les honeypots à faible interaction sont faciles à installer et à entretenir.
- Ils ne nécessitent pas de ressources informatiques importantes.
- Ils ne peuvent pas être compromis par des adversaires.

5.1.2. Inconvénients :

- Les honeypots à faible interaction n'offre qu'un accès limité au système d'exploitation.
- Ils sont facilement détectables par les attaquants en raison du manque de réponses attendues et de la mise en œuvre incomplète des services.

5.2. Les honeypots à moyenne interaction :

La caractéristique principale des honeypots à interaction moyenne est l'application virtualisation des couches. Ils ne sont pas destinés à simuler entièrement un environnement système opérationnel complet, ni à implémenter tous les détails d'un protocole d'application. Ce que font les honeypots d'interaction moyenne est de fournir des réponses suffisantes que les exploits connus attendent sur certains ports qui les inciteront à envoyer leurs charges utiles. Les honeypots les plus connus de ce type sont : homemade honeypots, Deception Toolkit. **[31]**

Le fonctionnement général de ce type de Honeypot est illustré par la figure 2.02.

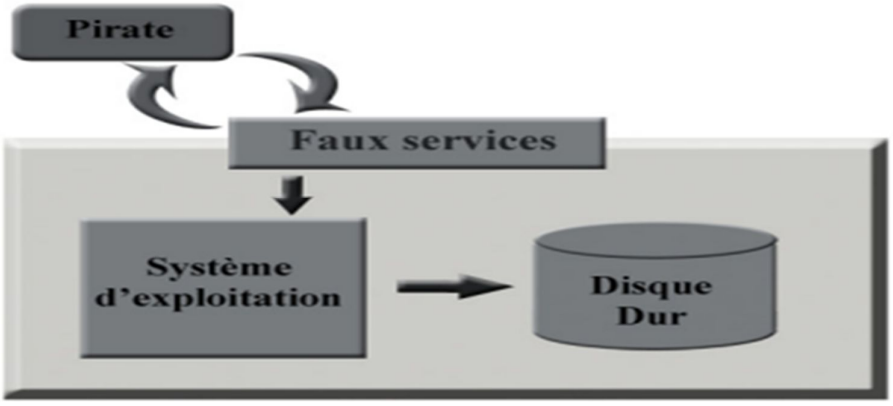

Figure 2.02: Schéma d'une interaction moyenne.

Les avantages et les inconvénients principaux de ce type de honeypots sont **[23]** :

5.2.1. Avantages :

- La gestion des logs du système est facile par rapport à celle des honeypots à haute interaction et un peu difficile par rapport à celle des honeypots à faible interaction.
- Fournit beaucoup plus d'informations intéressantes à analyser, à cause de variété d'attaques proposées aux pirates ce qui s'avère plus intéressant pour eux.

5.2.2. Inconvénients :

- Très dur à implémenter en terme de développement, car la fourniture d'un leurre parfait implique une parfaite connaissance des protocoles de chaque faux service pour bannir toute faille de sécurité.
- Sécurité du système difficile à contrôler, du fait que, plus le niveau de complexité d'un honeypot augmente plus il y a de chance qu'il contient lui-même un trou de sécurité qui peut être exploité par le pirate.

5.3. Les honeypots à haute interaction :

Les honeypots à haute interaction sont différents de leurs homologues à faible interaction dans la mesure où ils impliquent le déploiement de systèmes d'exploitation réels et d'applications avec lesquelles l'attaquant interagit. Les honeypots à haute interaction les plus connus sont : Honeynet CDROM ROO, ManTrap. **[32]**

Le fonctionnement général de ce type de Honeypot est illustré par la figure 2.03.

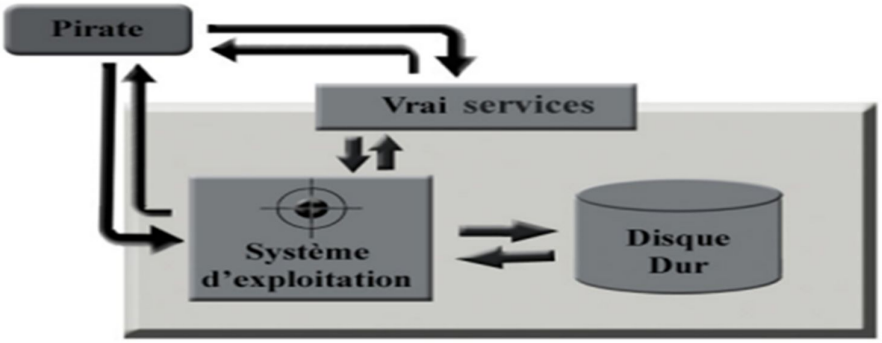

Figure 2.03: Schéma d'une interaction haute.

Les avantages et les inconvénients principaux de ce type de honeypots sont **[30]** :

5.3.1. Avantages :

- La possibilité de recueillir des informations détaillées sur les procédures d'un attaquant.
- L'absence de faux positifs est l'un des principaux avantages des honeypots à forte interaction par rapport aux systèmes de détection d'intrusion (IDS).

5.3.2. Inconvénients :

- L'attaquant peut accéder à un système informatique conventionnel et entreprendre des actions malveillantes.
- L'attaquant peut distinguer une machine virtuelle d'une machine réelle.

6. Architecture des honeypots :

L'architecture d'un honeypot est définie par la nature du système qui l'héberge, qui peut être réel ou virtuel **[33]** :

6.1. Architecture réelle :

Dans cette architecture, chaque honeypot est installé sur son propre appareil physique, c'est-à-dire que chaque honeypot est représenté par un système réel.

Cette architecture présente les avantages et les inconvénients suivants :

6.1.1. Avantages :

- Administration simplifiée.

6.1.2. Inconvénients :

- S'il y a plusieurs honeypots, plusieurs machines physiques.
- La surveillance du système sans être détectée par le hacker est complexe.
- Remise en place fréquente du système pour chaque honeypot.

6.2. Architecture virtuelle :

Dans cette architecture, le honeypot est installé sur une machine virtuelle. La création de machines virtuelles est garantie par des outils de virtualisation système, tels que : VMWare sous Linux et Windows, UML (User-Mode-Linux) sous Linux, Jail sous Unix BSD. Ces outils peuvent émuler un ou plusieurs systèmes sur une seule machine, de sorte que plusieurs honeypots virtuels peuvent être installés sur une seule machine. De plus, VMware peut émuler plusieurs systèmes de nature différente (Windows, linux, etc.) et fournir plusieurs honeypots de plusieurs systèmes d'exploitation virtuels sur la même machine physique.

Les avantages et les inconvénients de cette architecture sont résumés dans :

6.2.1. Avantages :

- Assurer la sécurité de la machine virtuelle.
- Economie de machines physiques.
- Capacité de contrôler les disques virtuels en temps réel.
- Réinstallation facile par la sauvegarde des disques virtuels.
- Le système d'hébergement des systèmes virtuels est rendu invisible au pirate.

6.2.2. Inconvénients :

- Chargement considérable sur le système hébergeant les machines virtuelles.
- Le choix du système virtuel est restreint à ceux qui sont compatibles.

7. Mise en place des honeypots :

Les honeypots peuvent être placés dans l'une des trois parties de l'organisme; ils peuvent être placés à l'extérieur sur Internet ou à l'intérieur de l'intranet. **[34]**

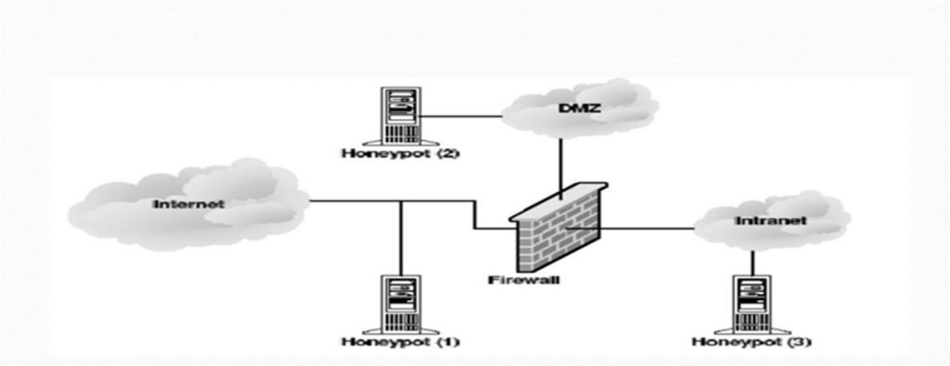

Figure 2.04: Placement du Honeypot.

7.1. Devant le pare-feu :

Les honeypots sont placés en dehors du périmètre du réseau quand on veut capturer les pirates les plus malveillants. Les honeypots connectés directement à Internet peuvent être librement compromis et sondés. C'est la configuration la plus facile pour une personne seule, à domicile et les honeypots de recherche. Le honeypot et le réseau de production partagent le même sous-réseau d'adresses IP publiques. Cela exige une ou plusieurs adresses IP publiques. S'il n'y a qu'une seule adresse IP publique et qu'un hub est disponible, l'adresse IP publique est donnée au honeypot et la station de surveillance est configurée sans adresse IP.

Le fait de placer des honeypots à l'extérieur du réseau réduit le risque pour le réseau interne, mais limite leur capacité d'émuler les systèmes de production et de générer des journaux, qui sont pertinents pour le réseau interne. **[29]**

Figure 2.05: Honeypot installé devant le pare-feu.

7.2. Derrière le pare-feu :

Un autre emplacement commun du système honeypot se trouve dans le réseau, avec le pare-feu entre lui et le monde extérieur. Ce placement est le meilleur moyen de créer un système d'alerte rapide pour générer une alerte si des exploits externes ont dépassé d'autres défenses de réseau et attraper des menaces internes en même temps**.** Par contre, si un honeypot intérieur est compromis, le contrôle des données au sein du réseau local est difficile. Un hacker ou un ver pourrait utiliser le honeypot exploité à la recherche d'hôtes internes supplémentaires à compromettre**. [29]**

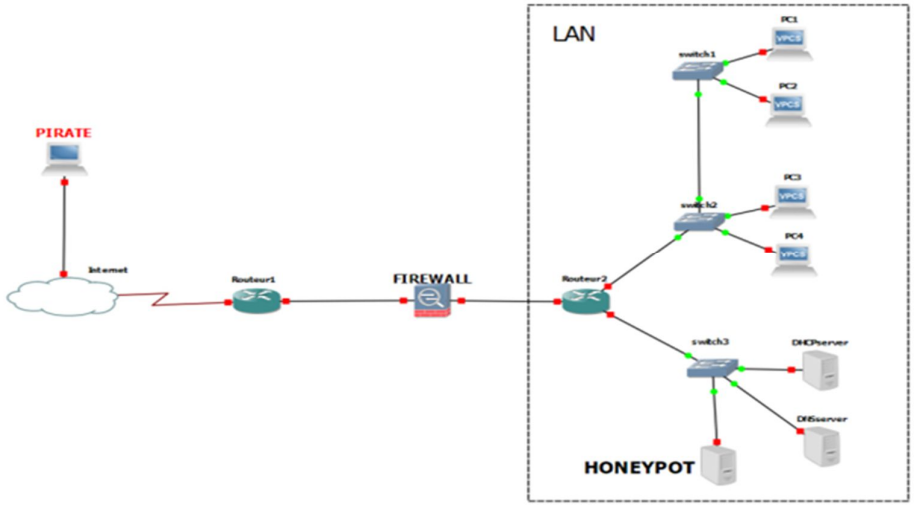

Figure 2.06: Honeypot installé derrière un pare-feu.

7.3. Dans une zone démilitarisée (DMZ) :

Mettre un honeypot sur la DMZ (zone démilitarisée) firewall est souvent le meilleur choix pour une entreprise. Il peut être placé aux côtés d'autres serveurs DMZ légitimes et fournir une alerte rapide des menaces sur ceux-ci. Un routeur est placé entre la DMZ du firewall comme couche supplémentaire pour le contrôle des données. Les serveurs honeynet et DMZ de production partagent le même schéma logique de sous-réseau et d'adresse IP. La DMZ peut être dotée d'adresses IP publiques ou privées. **[29]**

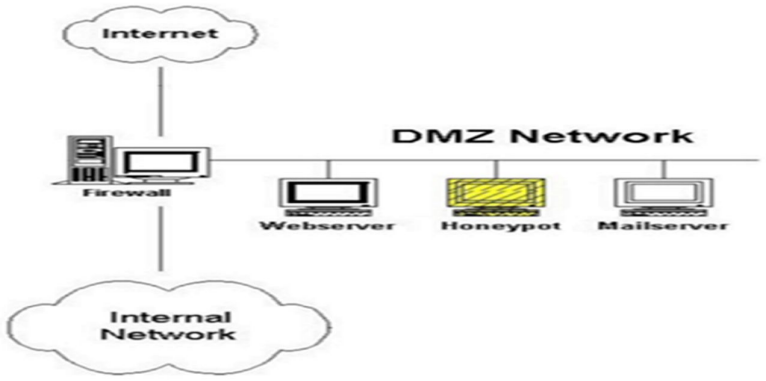

Figure 2.07: Honeypot installé dans une DMZ.

8. Avantages des honeypots :

Il y a de nombreux avantages des honeypots, mais nous concentrerons sur certains d'entre eux **[31]** :

8.1. Valeur de données:

L'un des défis auxquels fait face la collectivité de la sécurité consiste à tirer parti de la valeur des données. Les organisations collectent quotidiennement de grandes quantités de données, notamment des journaux de pare-feu, des journaux système et des alertes de détection d'intrusion. La quantité d'information peut être accablante, ce qui rend extrêmement difficile d'en tirer une valeur. Les honeypots, en revanche, collectent très peu de données, mais ce qu'ils collectent est généralement de grande valeur. Le concept de honeypot, qui ne permet aucune activité de production, réduit significativement le niveau de bruit. Au lieu d'enregistrer des giga-octets de données chaque jour, la plupart des honeypots collectent plusieurs méga-octets de données chaque jour, même si c'est beaucoup. Toute donnée enregistrée est probablement une analyse, une sonde ou une attaque ou information de grande valeur.

Les honeypots peuvent vous fournir les renseignements précis dont vous avez besoin dans un format rapide et facile à comprendre. L'analyse est ainsi beaucoup plus simple et le temps de réponse beaucoup plus rapide. Par exemple, le projet honeynet, un groupe de recherche sur les honeypots, recueille en moyenne moins de 1MB de données par jour. Même si c'est une très petite quantité de données, il contient surtout de l'activité malveillante. Ces données peuvent alors être utilisées pour modéliser les statistiques, analyser les tendances, détecter les attaques, voire même analyser les attaquants. Il ressemble à un effet microscopique. Les données que vous entrez sont passées au microscope pour un examen détaillé.

8.2. Ressources :

Un autre obstacle auquel sont confrontés la plupart des mécanismes de sécurité est le manque de ressources et même l'épuisement des ressources. L'épuisement des ressources est le point où une ressource de sécurité ne peut plus fonctionner car ses ressources sont dépassées. Par exemple, un pare-feu peut échouer car sa table de connexion est pleine, il ne dispose plus de ressources, ou il ne peut plus surveiller les connexions. Cela force le pare-feu à bloquer toutes les connexions au lieu de simplement bloquer les activités non autorisées. Un IDS peut avoir trop d'activité réseau à surveiller, voire des centaines de méga-octets de données par seconde. Quand cela se produit, les tampons du capteur IDS deviennent remplis, et il commence à déposer des paquets. Ses ressources ont été épuisées, et il ne peut plus surveiller efficacement l'activité du réseau, attaques éventuellement absentes. Les serveurs de journaux centralisés sont un autre exemple. Il se peut qu'ils ne puissent pas saisir tous les événements des systèmes à distance, qu'ils tombent ou qu'ils n'enregistrent pas les événements critiques. En revanche, la plupart des capteurs IDS ont des problèmes de surveillance des réseaux qui ont des vitesses gigabit. La vitesse et le volume de trafic sont tout simplement trop importants pour permettre au capteur d'analyser chaque paquet. Par conséquent, le trafic est éliminé et les attaques potentielles sont ratées. Un honeypot déployé au sein du même réseau ne partage pas ce problème. Il ne saisit que les activités dirigées contre lui-même, pour que le système ne soit pas submergé par le trafic. Lorsque le capteur IDS peut échouer en raison de l'épuisement des ressources, le honeypot n'est pas susceptible d'avoir un problème. Un avantage secondaire des besoins limités en ressources d'un honeypot est que vous n'avez pas à investir beaucoup d'argent dans le matériel pour elle. Contrairement à de nombreux mécanismes de sécurité tels que les pare-feu ou les capteurs IDS, les honeypots ne nécessitent pas la dernière technologie de pointe, de grandes quantités de RAM ou de puces, ou de gros disques durs.

8.3. Simplicité:

C'est le plus grand avantage des honeypots. Il n'y a aucun algorithme sophistiqué à développer, aucune base de données de signature à maintenir et aucune empreinte erronée. Prends le honeypot, mets-le quelque part dans ton organisation, et assieds-toi et attends. Alors que certains honeypot, en particulier les honeypots de recherche, peuvent être plus complexes, ils fonctionnent tous sur le même simple Prémisse : si quelqu'un se connecte au honeypot, Vérifiez-le. Comme vous le diront des spécialistes chevronnés de la sécurité, plus la conception est simple, plus elle est fiable. Avec la complexité viennent les erreurs, les pannes et les échecs.

8.4. Moins de faux positifs:

Toute interaction avec les honeypots sera considérée comme suspecte. De plus, lorsque toutes les personnes d'une organisation sont informées qu'il y a un honeypot dans l'organisation (c.-à-d. que certains appareils agissent comme des honeypots), personne n'essaiera d'y accéder.

8.5. Ne nécessitent pas de signatures d'attaque connues, contrairement à l'IDS:

Les honeypots ne nécessitent pas de signature d'attaque connue pour détecter les activités suspectes. Toutes les activités dans les honeypots seront stockées comme suspectes.

9. Désavantages des honeypots :

Il y a de nombreux désavantages des honeypots, mais nous nous concentrerons sur certains d'entre eux **[23]** :

9.1. Vision limitée:

Le plus grand inconvénient des honypots est qu'ils ont un champ de vision étroit : Ils voient seulement quelle activité est dirigée contre eux. Si un attaquant pénètre dans votre réseau et attaque une variété de systèmes, votre honeypots sera béatement ignorant de l'activité, sauf si elle est attaquée directement. Si l'attaquant a identifié votre honeypot pour ce qu'il est, elle peut maintenant éviter ce système et infiltrer votre organisation, avec le honeypot jamais savoir qu'elle est entrée. Comme nous l'avons déjà mentionné, les honeypots ont un effet microscopique sur la valeur des données que vous recueillez, ce qui permet vous concentrer étroitement sur les données de valeur connue. Cependant, comme un microscope, le champ de vision très limité du honeypot peut exclure les événements qui se produisent tout autour.

9.2. Prise d'empreinte :

Un autre inconvénient des honeypots, en particulier de nombreuses versions commerciales, est l'empreinte digitale. L'empreinte digitale est le moment où un attaquant peut identifier la véritable identité d'un honeypot parce qu'il a certaines caractéristiques ou comportements attendus. Un honeypot mal mis en œuvre peut également s'identifier. Par exemple, un honeypot peut être conçu pour émuler un serveur Web NT IIS, mais le honeypot à également certaines caractéristiques qui l'identifient comme un serveur Unix, Solaris. Ces identités contradictoires peuvent servir de signature pour un honeypot. Il y a une variété d'autres méthodes pour prendre des empreintes digitales d'un honeypot. La prise d'empreintes digitales est un risque encore plus grand pour les honeypots de la recherche. Un système conçu pour acquérir de l'intelligence peut être dévasté s'il

est détecté. Un attaquant peut fournir de mauvaises informations à un honeypot de recherche plutôt que d'éviter la détection. Cette mauvaise information amènerait alors les responsables de la sécurité à tirer des conclusions erronées au sujet de la communauté des casques noirs.

9.3. Risque :

Le troisième défaut des honeypots est le risque : ils peuvent présenter un risque pour votre environnement. Par risque, nous voulons dire qu'un honeypot, une fois attaqué, peut être utilisé pour attaquer, infiltrer ou blesser d'autres systèmes ou organisations. Plus tard, différents honeypots ont différents niveaux de risque. Certains introduisent très peu de risques, alors que d'autres donnent à l'attaquant des plates-formes complètes à partir desquelles lancer de nouvelles attaques. Plus le honeypot est simple, moins le risque est grand. Par exemple, un honeypot qui ne fait que reproduire quelques services est difficile à compromettre et à utiliser pour attaquer d'autres systèmes. D'autre part, un honeypot qui crée une prison donne à un agresseur un système d'exploitation réel avec lequel interagir. Un attaquant pourrait être capable de sortir de cette cage et ensuite utiliser le honeypot pour lancer des attaques passives ou actives contre d'autres systèmes ou organisations. Le risque varie selon la conception et le déploiement du pot à miel. En raison de leurs inconvénients, les honeypots ne peuvent pas remplacer d'autres mécanismes de sécurité tels que les pare-feu et les systèmes de détection d'intrusion. Ils ajoutent plutôt de la valeur en travaillant avec les mécanismes de sécurité existants. Ils jouent un rôle dans vos défenses globales.

10. Exemples des honeypots :

10.1. BackOfficer Friendly :

BackOfficer Friendly est un honeypot simple à faible interaction, développé par Marcus Ranum, fonctionne dans un environnement graphique (Windows ou Unix), il émule quelques services de base comme : http, ftp, Telnet, mail, ou Back Orifice3. Cependant, il vous permet d'activer quelques réponses pour certains services (en utilisant l'option "réponses factices"), des réponses limitées et non configurables. Le programme ressemble également à un système de détection d'intrusion simple qui enregistre sous forme d'une liste d'avertissements toutes les tentatives de connexion aux ports qu'il contrôle. Si cela ne présente que peu d'intérêt pour un utilisateur avancé, les utilisateurs non formés trouveront une très bonne solution pour se familiariser avec les technologies honeypot. **[35]**

10.2. Spectre :

Les auteurs de Spectre décrivent Spectre comme un "système de détection d'intrusion basé sur un honeypot". Cependant, le produit est principalement un honeypot conçu pour éloigner les attaquants des systèmes de production et collecter des preuves contre les attaquants. Spectre a quelques fonctionnalités intéressantes que l'on ne trouve pas dans d'autres solutions :

- Spectre rend les données leurres disponibles pour que les attaquants puissent y accéder et les télécharger. Ces fichiers de données laissent des traces sur l'ordinateur de l'attaquant comme preuve.
- Spectre peut émuler des machines dans différents états : un système mal configuré, un système sécurisé, un système défaillant système (avec des pannes matérielles ou logicielles) ou un système imprévisible.
- Spectre tente activement de collecter des informations sur chaque attaquant. **[36]**

10.3. Honeyd :

Honeyd est un honeypot très complet, facile à utiliser et à faible interaction élaboré par Niels Provos de l'Université du Michigan. Il fonctionne sur les systèmes Unix et est porté sur Windows. Il simule les services et même les vrais systèmes d'exploitation sur les adresses IP inutilisées sur un réseau. Grâce aux fichiers de configuration, Honeyd peut émuler de nombreux services de personnaliser les réponses aux connexions, de simuler différentes piles IP pour tromper l'attaquant sur la version du système d'exploitation. **[33]**

10.4. Nepenthes :

L'idée principale derrière les nepenthes est l'émulation des vulnérabilités du réseau prestations de service. Au lieu de déployer un honeypot avec une forte interaction avec les services qui peuvent être exploités par des logiciels malveillants indépendants, ce programme émule juste les services. D'une part, cela réduit le risque de courir un filet de miel. Puisque nepenthes n'effectue pas un service vulnérable, un attaquant ne peut pas compromettre complètement votre honeypot. Le processus d'attaque interagira avec l'émulation, de sorte que nous atténuons le risque. Lorsque nous avons téléchargé un logiciel malveillant, il est stocké sur le disque dur et ne fonctionne jamais. Même si cela serait exécuté, il est très peu probable que le binaire serait exécuté parce qu'il cible un système Windows, mais nepenthes fonctionne sous Linux. Ainsi, le honeypot n'est jamais infecté par des malwares - ce qui est impossible avec un honeypot à forte interaction ou d'autres approches.

En revanche, cette méthodologie mène à une meilleure extensibilité. Les honeypots à faible interaction ont l'avantage de pouvoir gérer plusieurs milliers de honeypots sur une seule machine physique. **[30]**

10.5. Déception Toolkit (DTK) :

Le DTK a la capacité d'imiter de manière trompeuse les systèmes d'exploitation suivants : Windows NT, Linux, HP-UX, SCO Unix, SGI, IBM AIX, Sun Solaris, SunOS et Ultrix. La boîte à outils de tromperie a été utilisée comme architecture principale du honeypot. Premièrement, il a été conçu pour être utilisé comme un outil défensif que les administrateurs système pourraient utiliser pour défendre les systèmes. Le DTK imite les serveurs standards de l'industrie et les services qu'ils fournissent en écoutant les entrées et en redirigeant le trafic vers PERL personnalisable fichiers de scripts. Ces fichiers de script répondent alors comme un serveur ou un démon installé devrait le faire lorsqu'ils sont envoyés commandes légitimes ou non, sélectionnées en déployant de faux services, la machine semblera contenir de nombreux ports utiles, et il affichera des réponses destinées à apparaître typiques d'un serveur en fonctionnement. En faisant cela, le DTK enregistrera les actions de l'intrus par le biais de fichiers journaux qui pourront ensuite être analysés pour déterminer le mode opératoire de l'agresseur. Le honeypot trompeur est destiné à prolonger le temps de détection fenêtre alors que l'attaquant est entraîné dans des services de sondage qui sont des caméléons numériques qui n'ont pas de véritable charge utile ou substance. **[37]**

10.6. ManTrap :

Mantrap est un produit très complet proposé par Symantec. Ce honeypot est très interactif et met en œuvre quatre systèmes d'exploitation logiquement séparés qui sont installés sur la même machine hôte. Chacun de ces systèmes prend en charge des applications réelles, et apparaît en tant que système autonome avec sa propre interface réseau. Le système hôte est administré à travers une interface utilisateur graphique Java. Mantrap peut être utilisé comme honeypot de production, particulièrement pour la détection et la réaction, et il est également rentable dans la recherche, mais avec un risque considérable. **[35]**

11. La différence entre honeypot et IDS, IPS, Firewall :

Honeypot	IDS	IPS	Firewall
Le honeypot est un système de Prévention et Détection et Réponse.	L'IDS est un système de surveillance.	L'IPS est un système de contrôle.	Un firewall est un outil informatique (matériel et/ou logiciel) conçu pour protéger les données d'un réseau.
Le honeypot est un système rendu volontairement vulnérable afin d'attirer les attaquants, observer leurs techniques et récupérer leurs outils.	Ils assurent principalement la détection des techniques de sondage, des tentatives de compromission de systèmes, d'activités suspectes internes ou des activités virales.	La fonction principale est d'empêcher toute activité suspecte détectée au sein d'un système.	Le firewall effectue des actions telles que le blocage et le filtrage du trafic.
Selon que le système leurre est destiné à surveiller les attaques extérieures ou bien internes au réseau de l'organisation, il existe trois positions possibles pour installer un honeypot : Devant le pare-feu, dans une zone démilitarisé (DMZ) ou derrière le pare-feu.	L'IDS est placé à la périphérie d'un réseau pour collecter tous les événements, enregistrer et détecter les violations.	L'IPS est placé derrière le pare-feu du réseau et communique en ligne avec le trafic entrant pour mieux prévenir les intrusions.	Il repose parfois sur un matériel réseau dédié, constituant un intermédiaire entre le réseau local (ou la machine locale) et un ou plusieurs réseaux externes.
Il existe deux types de classification : Une première qui consiste à les classer selon les interactions qu'ils permettent, et une seconde classification qui les catégorise selon leur intérêt d'utilisation.	L'IDS est divisé en fonction de l'endroit où se produit la détection et de la menace ou de la méthode de détection utilisée.	l'IPS sont divisés en différents types selon leur fonctionnalité comme Network based IPS, Host Based IPS.	Deux types de firewall existent : le pare-feu matériel et le pare-feu logiciel. En fonction de la situation, il est possible d'installer l'un ou l'autre, ou de cumuler les deux pour accroître la sécurité du réseau.

Tableau 2.01: La différence entre honeypot, IDS, IPS et Firewall.

12. Conclusion :

Dans ce chapitre, nous avons parlé des honeypots, commençant par la définition et les objectives. Ensuite, les différents types des honeypots classifiés selon leur intérêt d'utilisation et les interactions qu'ils permettent**.** Nous avons parlé de la mise en place des honeypots aussi ; ainsi que les avantages et les limites de ce dernier et nous avons clôturé le chapitrer par présenter quelques types des honeypots.

Dans le prochain chapitre, On va présenter brièvement le projet, Les ressources matérielles et logiciels. Après le honeypot MHN et ses fonctionnalistes et les types des honeypots pris en charge par ce dernier. Dans ce projet, nous avons utilisé le honeypot « cowrie », nous avons commencé

tous d'abord par une définition et quelque fonctionnalité de cet outil, ensuite, l'installation et la configuration du serveur MHN et du honeypot.

1. Présentation du projet :

L'objectif principal de ce mémoire consiste à identifier les attaques et les ressources qui intéressent les pirates informatiques afin de mieux sécuriser les infrastructures et les données dans un réseau. Le honeypot permettra aussi de récolter des informations sur les pirates responsables des attaques. Afin de mettre en œuvre tout ce processus. Ce mémoire proposera une implémentation d'une honeypot et une simulation des attaques couramment observés.

La figure 3.01 représente l'architecture du réseau :

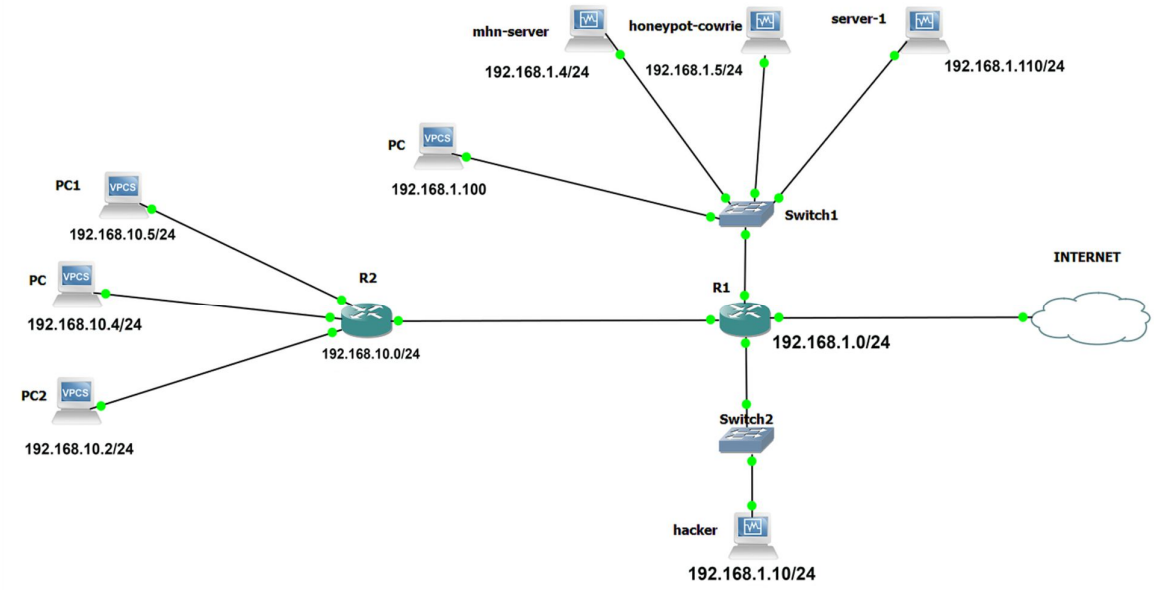

Figure 3.01: Architecture du réseau.

2. Les ressources utilisées :

2.1. Ressources matérielles :

La seule ressource matérielle utilisée dans ce projet est un ordinateur.

La configuration minimale requise est la suivante :

- RAM : 8 Go ou plus.
- Processeur : CORE i5 ou plus.
- 150 GB d'espace libre ou plus.

2.2. Ressources logicielles :

L'élément principal est d'abord un système d'exploitation. L'OS utilisé ici est Windows 10 mais il est également possible d'utiliser des distributions Linux ou Mac Os.

Oracle VM VirtualBox : Il permet de créer des machines virtuelles et de les exécuter en même temps au-dessus d'un système d'exploitation hôte. Nous devons créer trois machines virtuelles contenant les systèmes d'exploitation UBUNTU et KALI Linux. La première machine contient le serveur et le système d'exploitation Ubuntu. La deuxième machine contient le honeypot et le système d'exploitation Ubuntu. La troisième machine contient le système d'exploitation KALI LINUX elle sera utilisé pour simuler les attaques informatiques.

Les outils de travail		
Nom de la machine	**Système d'exploitation**	Outils et services
MHN-Server	Linux-ubuntu 18.04	Gestion des honeypots
honeypot-Cowrie	Linux-ubuntu 18.04	SSH-Honeypot (Cowrie)
hacker	Linux-Debian	NMAP

Tableau 3.01: Les outils de travail.

3. Le Honeypot MHN (odern Honey Network):

Il s'agit d'un serveur central open source pour la gestion et la collecte des données à partir de honeypots, Il a été développé par "Threat Stream" en 2015 et comporte des capteurs faciles à déployer rapidement Les données sont immédiatement collectées et affichées dans l'interface web. **[38]**

3.1. Fonctionnalités MHN :

- Il permet aux organisations de créer des Honeypot en quelques minutes.
- Facilité de déploiement de plusieurs Honeypot sur le même serveur.
- Gérez et collectez facilement les informations des Honeypots et affichez-les dans une interface graphique.
- Fournir des résultats différents et variés sur les attaquants et leurs méthodes et ainsi savoir ce qu'ils essaient de réaliser pour le protéger.
- MHN est open source sous la licence GNU GPLV3.

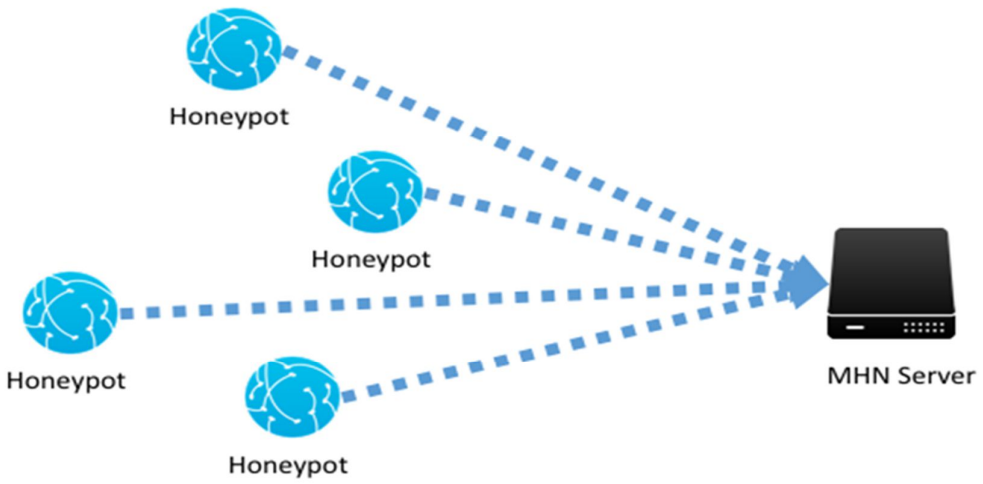

Figure 3.02: Serveur central MHN gérant et collectant les données.

3.2. Types de honeypots pris en charge par MHN :

- Ubuntu - Wordpot - Un émulateur WordPress.
- Ubuntu - p0f - Empreinte digitale du système d'exploitation passif.
- Ubuntu - Shockpot - Détecte les tentatives d'exploitation ShellShock.
- Ubuntu - cowrie - SSH et Telnet honeypot.
- Ubuntu - Suricata - Système de détection d'intrusion.
- Raspberry Pi - Dionaea - Honeypot capturant des logiciels malveillants.
- Redhat/Centos - Kippo - SSH honeypot.
- Ubuntu - Kippo as vulnerable Juniper Netscreen.
- Ubuntu - Conpot - Ihoneypot des systèmes de contrôle industriel.
- Ubuntu/Raspberry Pi - Kippo - SSH honeypot.
- Ubuntu - ElasticHoney - Honeypot Elasticsearch.
- Ubuntu - Amun - Python honeypot.
- Ubuntu - Dionaea with HTTP - Honeypot capturant des logiciels malveillants.
- Ubuntu - Snort - ISystème de détection d'intrusion.
- Ubuntu - Dionaea - Honeypot capturant des logiciels malveillants.
- Ubuntu - Shockpot Sinkhole - Détection de shellshock et gouffre.

3.3. Le type de honeypot utilisé dans le projet :

- ✓ Nous choisirons (Cowrie)-SSH-Honeypot pour tester l'attaque « Brute-force».

3.3.1. Cowrie :

C'est un outil écrit en python qui simule l'intégralité du serveur ssh, et l'outil peut interagir avec l'utilisateur et continuer l'astuce avec lui avant l'attaque et après l'attaque afin qu'il essaie de casser le processus de connexion ssh et si l'administrateur du serveur veut qu'il réussisse, il facilitera le mot de passe, puis lui permettra d'entrer dans un système de fichiers fictif contenant une copie exacte d'un système de fichiers sur un serveur réel pouvant entrer dans le dossier « etc » extraire le fichier de mot de passe et faire ce qu'il veut créer des fichiers, supprimer et beaucoup de choses avec tout cela, l'administrateur du serveur saura ce que le pirate veut du serveur et quels fichiers et type d'informations qu'il veut retirer et quel est le but du hack et beaucoup d'information. **[39]**

3.3.2. Certaines des fonctionnalités de l'outil Cowrie :

- Un faux système de fichiers semblable à un vrai système de fichiers qui permet au pirate de créer, d'ajouter, de supprimer et de retirer ce qu'il veut du système et de parcourir des chemins comme ceux du système réel.
- Ajouter faux contenu dans le système de fichiers de l'outil par exemple, il est possible de mettre des fichiers sur le dossier Accueil et d'autres dossiers, en ajoutant des utilisateurs et bien d'autres choses qui font qu'on se sent à l'intérieur d'un vrai système de fichiers.
- Si l'attaquant télécharge des fichiers sur l'outil, il les conservera afin que l'administrateur puisse lire et identifier.

4. Installation et la configuration du serveur MHN et du honeypot :

4.1. Installation et la configuration du MHN-serveur :

- ➢ Au début il faut installer l'outil git qui nous permettra d'installer mhn dans notre machine virtuel du serveur.

```
server@server-VirtualBox:/$  sudo apt install git -y
```

- Pour installer le serveur mhn, vous devez exécuter les trois commandes suivantes :

  - ✓ Ouvrir le dossier opt.

```
server@server-VirtualBox:/$  cd / opt /

server@server-VirtualBox:/opt $
```

  - ✓ Copiez le dossier mhn qui se trouve dans le github.

```
server@server-VirtualBox:/opt $  sudo git clone
https://github.com/threatstream/mhn.git
```

  - ✓ Ouvrir le dossier mhn.

```
server@server-VirtualBox:/opt $ cd mhn/

server@server-VirtualBox:/opt / mhn $
```

- Dans le dossier mhn exécute le fichier 'install.sh' pour lancer l'installation

```
server@server-VirtualBox:/opt / mhn $ sudo./install.sh
```

- Afin de terminer l'installation de mhn il doit être configuré.

```
+ echo ' MHN Configuration ' MHN Configuration + echo
==========================================================
========================================================== +
python generateconfig.py

 Do you wish to run in Debug mode?: y/n n

Superuser email: chindje21@gmailcom

Superuser password:

 Superuser password: (again):

 Server base url ["http:// 105.110.58.177 "]: http://192.168.1.4

Honeymap url ["http:// 105.110.58.177:3000"]: http://192.168.1.4:300
```

Mail server address ["localhost"]:

Mail server port [25]:

Use TLS for email ?: y /n n

Use SSL for email ?: y /n n

[“ “] Mail server username :

[“ “] Mail server password :

[“ “] Mail default sender :

Patch for log file [“/var/log/mhn/mhn.log”]:

+echo –e ‘\nlnitializing database,please be patient.This can take serveral minutes’

lnitializing database,please be patient.This can take serval minutes

+python initdatabase.py

- Après avoir configuré mhn, vérifiez s'il fonctionne correctement.
  - ✓ Nginx.

```
server@server-VirtualBox:/opt/mhn$ sudo /etc/init.d/nginx status
[sudo] password for server:
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: e
nabled)
   Active: active (running) since Wed 2022-06-15 15:51:08 CEST; 11min ago
     Docs: man:nginx(8)
  Process: 836 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code
=exited, status=0/SUCCESS)
  Process: 727 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process
on; (code=exited, status=0/SUCCESS)
 Main PID: 837 (nginx)
    Tasks: 3 (limit: 3474)
   CGroup: /system.slice/nginx.service
           ├─837 nginx: master process /usr/sbin/nginx -g daemon on; master_…n;
           ├─838 nginx: worker process
           └─839 nginx: worker process

juin 15 15:51:00 server-VirtualBox systemd[1]: Starting A high performance …....
juin 15 15:51:08 server-VirtualBox systemd[1]: Started A high performance w…er.
Hint: Some lines were ellipsized, use -l to show in full.
server@server-VirtualBox:/opt/mhn$
```

Figure 3.03: L’état du service Nginx.

✓ Unix.

```
   Active: active (running) since Wed 2022-06-15 15:51:00 CEST; 11min ago
     Docs: http://supervisord.org
 Main PID: 726 (supervisord)
    Tasks: 13 (limit: 3474)
   CGroup: /system.slice/supervisor.service
           ├─ 726 /usr/bin/python /usr/bin/supervisord -n -c /etc/supervisor…nf
           ├─ 919 /opt/mhn/env/bin/python /opt/mhn/env/bin/celery beat -A mh…FO
           ├─ 922 /opt/hpfeeds/env/bin/python /opt/hpfeeds/examples/geoloc/g…on
           ├─ 923 /opt/mhn/env/bin/uwsgi -s /tmp/uwsgi.sock -w mhn:mhn -H /o…60
           ├─ 924 /opt/mhn/env/bin/python /opt/mhn/env/bin/celery worker -A …FO
           ├─ 925 /opt/mhn/env/bin/python collector_v2.py collector.json
           ├─ 926 /opt/honeymap/server/server
           ├─1015 /opt/mhn/env/bin/python /opt/mhn/env/bin/celery worker -A …FO
           └─1016 /opt/mhn/env/bin/python /opt/mhn/env/bin/celery worker -A …FO

juin 15 15:51:33 server-VirtualBox supervisord[726]: 2022-06-15 15:51:33,867…d)
juin 15 15:51:34 server-VirtualBox supervisord[726]: 2022-06-15 15:51:34,077…d)
juin 15 15:51:35 server-VirtualBox supervisord[726]: 2022-06-15 15:51:35,078…ly
juin 15 15:51:35 server-VirtualBox supervisord[726]: 2022-06-15 15:51:35,079…79
juin 15 15:51:35 server-VirtualBox supervisord[726]: 2022-06-15 15:51:35,324…d)
juin 15 15:51:37 server-VirtualBox supervisord[726]: 2022-06-15 15:51:37,797…89
juin 15 15:51:38 server-VirtualBox supervisord[726]: 2022-06-15 15:51:38,014…d)
juin 15 15:51:41 server-VirtualBox supervisord[726]: 2022-06-15 15:51:41,288…09
juin 15 15:51:41 server-VirtualBox supervisord[726]: 2022-06-15 15:51:41,525…d)
juin 15 15:51:41 server-VirtualBox supervisord[726]: 2022-06-15 15:51:41,800…ly
Hint: Some lines were ellipsized, use -l to show in full.
server@server-VirtualBox:/opt/mhn$
```

Figure 3.04: L'état du service Unix.

➢ Nous nous connectons à l'interface MHN-Server en tapant localhost ou l'adresse IP privée du serveur dans le moteur de recherche et en entrant E-mail et le mot de passe.

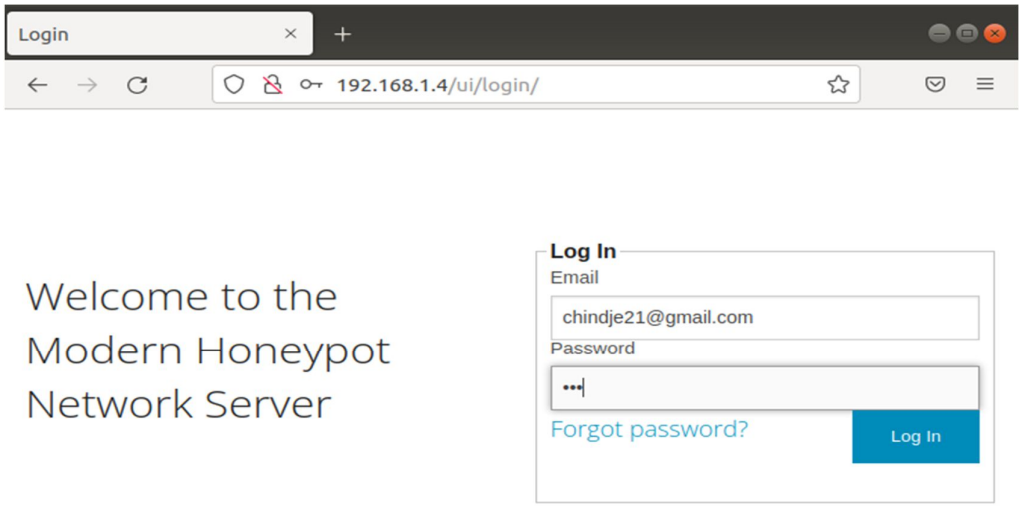

Figure 3.05: Page login du serveur MHN.

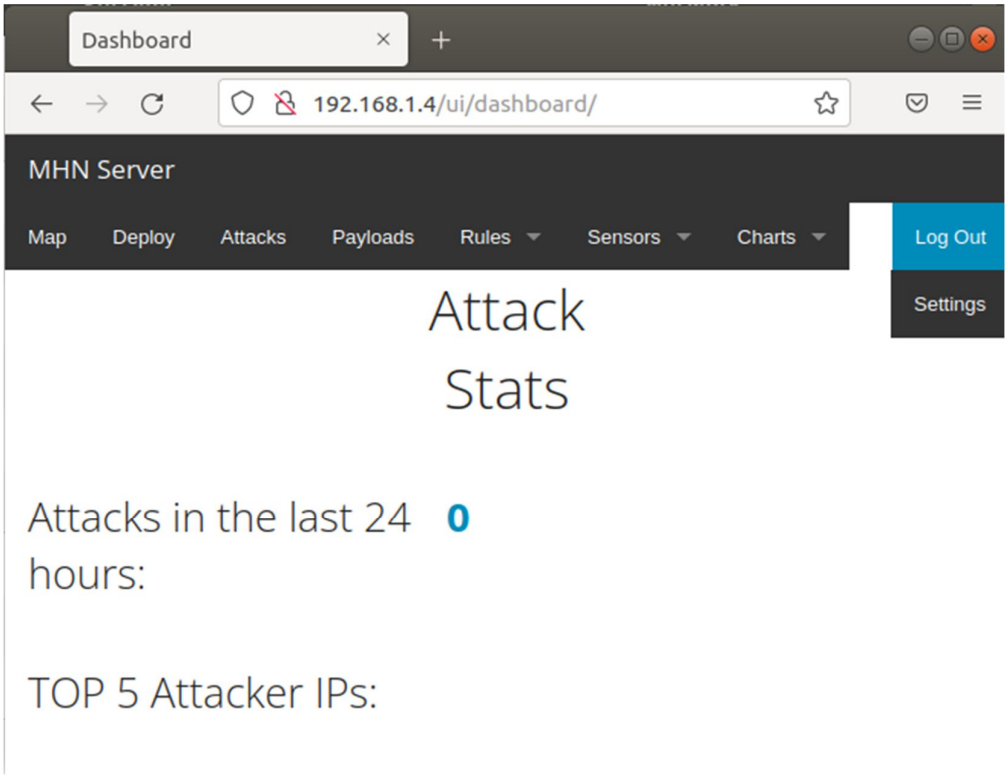

Figure 3.06: La page d'accueil.

4.2. Installation du honeypot-cowrie :

- En reste tout jour dans la machine virtuelle du serveur, dans la liste des options en haut, nous choisissons le type de honeypot à utiliser.

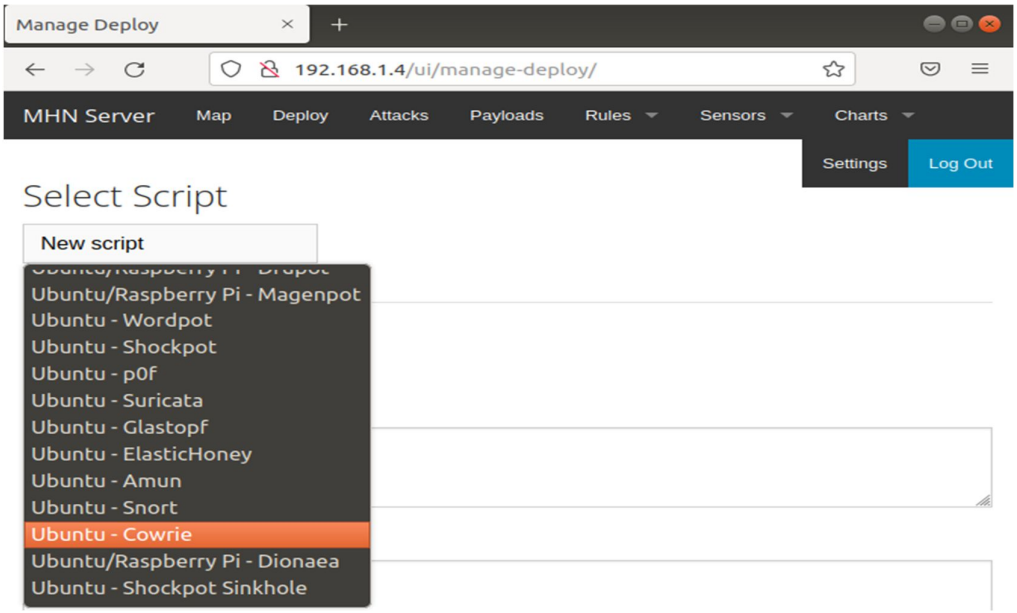

Figure 3.07: La page du déploiement.

- Après avoir sélectionné le type de Honeypot, le script suivant apparaîtra, que nous exécutons dans la machine virtuelle du honeypot.

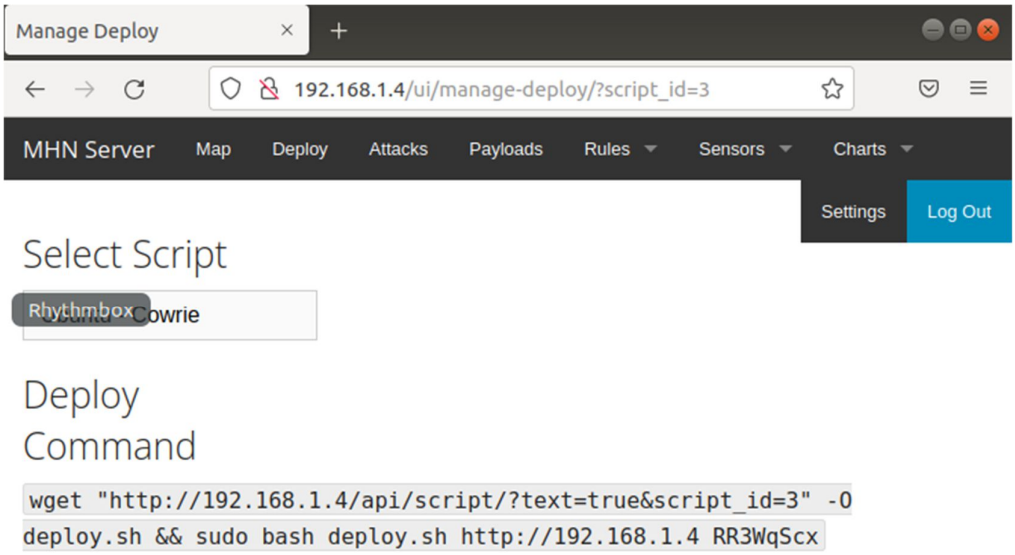

Figure 3.08: Script du honeypot-cowrie.

- Nous nous connectons à la machine virtuelle du honeypot, puis ce script téléchargera le type de honeypot qui a été sélectionné (cowrie).

```
pot-de@potde-VirtualBox:~$ sudo wget "http://192.168.1.4/api/script/?text=true&
script_id=3" -O deploy.sh && sudo bash deploy.sh http://192.168.1.4 RR3WqScx
[sudo] Mot de passe de pot-de :
--2022-06-16 00:03:48--  http://192.168.1.4/api/script/?text=true&script_id=3
Connexion à 192.168.1.4:80… connecté.
requête HTTP transmise, en attente de la réponse… 200 OK
Taille : 3166 (3,1K) [text/html]
Enregistre : «deploy.sh»

deploy.sh           100%[==================>]   3,09K  --.-KB/s    ds 0s

2022-06-16 00:03:48 (402 MB/s) - «deploy.sh» enregistré [3166/3166]

+ '[' 2 -ne 2 ']'
+ apt-get update
Réception de :1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88
,7 kB]
Atteint :2 http://fr.archive.ubuntu.com/ubuntu bionic InRelease
Réception de :3 http://fr.archive.ubuntu.com/ubuntu bionic-updates InRelease [8
8,7 kB]
Réception de :4 http://fr.archive.ubuntu.com/ubuntu bionic-backports InRelease
[74,6 kB]
Réception de :5 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Pa
ckages [2 304 kB]
Réception de :6 http://fr.archive.ubuntu.com/ubuntu bionic-updates/main amd64 P
ackages [2 646 kB]
Réception de :7 http://security.ubuntu.com/ubuntu bionic-security/main i386 Pac
kages [1 198 kB]
```

Figure 3.09: Installation de honeypot.

- Pendant le processus de téléchargement, le vrai port 22 sera converti en port 2222 et le faux service SSH fonctionnera sur le port 22.

```
supervisor est déjà la version la plus récente (3.3.1-1.1).
git est déjà la version la plus récente (1:2.17.1-1ubuntu0.11).
libssl-dev est déjà la version la plus récente (1.1.1-1ubuntu2.1~18.04.17).
openssl est déjà la version la plus récente (1.1.1-1ubuntu2.1~18.04.17).
python-pip est déjà la version la plus récente (9.0.1-2.3~ubuntu1.18.04.5).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 118 non mis à jour.
+ pip install -U supervisor
The directory '/home/pot-de/.cache/pip/http' or its parent directory is not own
ed by the current user and the cache has been disabled. Please check the permis
sions and owner of that directory. If executing pip with sudo, you may want sud
o's -H flag.
The directory '/home/pot-de/.cache/pip' or its parent directory is not owned by
 the current user and caching wheels has been disabled. check the permissions a
nd owner of that directory. If executing pip with sudo, you may want sudo's -H
flag.
Requirement already up-to-date: supervisor in /usr/local/lib/python2.7/dist-pac
kages
Requirement already up-to-date: setuptools in /usr/local/lib/python2.7/dist-pac
kages (from supervisor)
+ /etc/init.d/supervisor start
[ ok ] Starting supervisor (via systemctl): supervisor.service.
+ sed -i s/#Port/Port/g /etc/ssh/sshd_config
+ sed -i 's/Port 22$/Port 2222/g' /etc/ssh/sshd_config
+ service ssh restart
+ useradd -d /home/cowrie -s /bin/bash -m cowrie -g users
```

Figure 3.10: Changement du port SSH.

- Maintenant, après le processus d'installation de cowrie, nous vérifions s'il fonctionne correctement.

```
honeypot-cowrie@honeypotcowrie-virtualBox:~$ sudo supervisorctl status
cowrie                           RUNNING    pid 19218, uptime 0:01:24
honeypot-cowrie@honeypotcowrie-virtualBox:~$
```

- En entrant dans le MHN-serveur et en entrant dans le menu Sensors, il nous apparaît clairement que le processus de connexion a été établi entre MHN-Server et SSH-Honeypot.

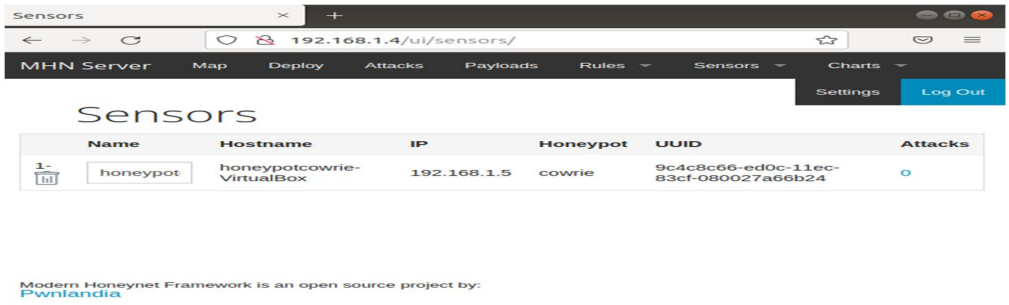

Figure 3.11: La page sensors.

5. Conclusion :

En guise de conclusion, le déploiement d'un Honeypot dans notre réseau est une tâche qui doit être effectuée minutieusement afin d'en tirer le maximum d'information possible sur l'attaquant. Des négligences au niveau des configurations pourraient entraîner des conséquences désastreuses. Maintenant, que tout est bien configuré, le chapitre suivant se chargera de tester l'ensemble à partir d'une simulation consistant émuler une attaque dans notre réseau.

1. Mise en œuvre de la simulation :

1.1. Identification des vulnérabilités :

La machine de l'attaquant qui est sous Kali linux (Debian) est un OS spécialisé pour les tests de pénétration réseau. Il dispose donc de plusieurs outils permettant de scanner les vulnérabilités mais aussi de les exploiter.

L'attaquant attaque le réseau en suivant les étapes suivantes :

- Network Scanning : en ligne ou hors ligne.
- Port Scanning : qu'est-ce que le service sur les ports ?
- Service Scanning : attaquez n'importe quel service.

➢ Maintenant, via le système hacker-hacker, nous allons installer l'outil Nmap :

```
hacker@hacker:~$ sudo apt install nmap
[sudo] password for hacker:
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

➢ Scanner le réseau dans lequel se trouvent les serveurs 198.168.1.0/24

La commande Nmap -sP : elle enverra un ensemble de requête ping pour déterminer quels appareils sont ouverts ou fermés.

```
hacker@hacker:~$ nmap -sP 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-16 16:20 CEST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Try using --system-dns or specify valid servers with --dns-servers
 Nmap scan report for 192.168.1.4
```

```
Host is up (0.022s latency).

Nmap scan report for 192.168.1.5

Host is up (0.013s latency).

Nmap scan report for 192.168.1.110

Host is up (0.24s latency).

Nmap scan report for 192.168.1.100

Host is up (0.60s latency).
```

- Maintenant vérifiez les ports de tous les appareils avec lesquels la communication est disponible En ligne.

```
kali@attacker:~$ sudo nmap -sS -A 192.168.1.4

Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-16 16:22 CEST

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is
disabled.

Try using --system-dns or specify valid servers with --dns-servers

 Nmap scan report for 192.168.1.4 Host is up (0.060s latency).

All 1000 scanned ports on 192.168.1.4 are closed

Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port

Aggressive OS guesses: Cisco 1812, 3640, or 3700 router (IOS 12.4) (95%), Cisco
DOCSIS cable modem termination server (IOS 12.1) (95%), Cisco Catalyst 3560 or
6500-series switch (IOS 12.1 - 12.2) (95%), Cisco ASR 1002 router (94%), Cisco
SOHO 97 ADSL router (94%), Cisco uBR10012 broadband router (94%), Cisco
1841 router (IOS 12) (94%), Cisco 1841 router (IOS 12.4) (94%), Cisco 2801 router
(IOS 12.4) (94%), Cisco 7600 router (IOS 12.2) (94%)

 No exact OS matches for host (test conditions non-ideal).

 Network Distance: 1 hops

TRACEROUTE (using port 199/tcp)

HOP RTT
```

ADDRESS 1 6.85 ms 192.168.1.4

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.

Nmap done: 1 IP address (1 host up) scanned in 9.15 seconds

kali@attacker:~$

- La première machine : Tous les 1000 ports scannés sur 192.168.1.4 sont fermés

kali@attacker:~$ sudo nmap -sS -A 192.168.1.100

Starting Nmap 7.80 (https://nmap.org) at 2022-06-16 16:24 CEST

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.

Try using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 192.168.1.100

Host is up (0.019s latency).

All 1000 scanned ports on 192.168.1.100 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 1 hops

TRACEROUTE (using port 8888/tcp)

HOP RTT

ADDRESS

1 3.67 ms 192.168.1.100

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.

Nmap done: 1 IP address (1 host up) scanned in 12.24 seconds

kali@attacker:~$

- La deuxième machine : Tous les 1000 ports scannés sur 192.168.1.100 sont fermés.

```
kali@attacker:~$ sudo nmap -sS -A 192.168.1.110

Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-16 16:26 CEST

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.

Try using --system-dns or specify valid servers with --dns-servers

Nmap scan report for 192.168.1.110

Host is up (0.019s latency).

All 1000 scanned ports on 192.168.1.110 are closed

Too many fingerprints match this host to give specific OS details

Network Distance: 1 hops

TRACEROUTE (using port 8888/tcp)

HOP RTT

ADDRESS

1 2.67 ms 192.168.1.110

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/.

Nmap done: 1 IP address (1 host up) scanned in 12.24 seconds

kali@attacker:~$
```

- La troisième machine: Tous les 1000 ports scannés sur 192.168.1.110 sont fermés.

Remarque : La commande nmap -sS -A est considérée comme l'un des types les meilleurs et les plus largement utilisés dans le processus de vérification des ports.

```
sudo nmap -sS -A 192.168.1.5

Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-16 16:50 CEST

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.

Try using --system-dns or specify valid servers with --dns-servers
```

```
Nmap scan report for 192.168.1.5
Host is up (0.039s latency).
Not shown: 999 closed ports
PORT STATE SERVICE VERSION
22/tcp
open
ssh OpenSSH 6.7p1 Ubuntu 5ubuntu1.3 (Ubuntu Linux)
Network Distance: 1 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
TRACEROUTE (using port 587/tcp)
HOP RTT
ADDRESS
1 3.95 ms 192.168.1.5
OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 40.64 seconds
```

Le service SSH-Honeypot a répondu à l'attaquant pour tromper qu'il dispose d'un service SSH disponible pour se connecter, mais un faux qui attire l'attaquant pour l'éloigner des systèmes importants, et dès la détection nous prenons immédiatement les mesures nécessaires pour l'empêcher d'essayer de se connecter à nouveau au réseau et une adresse IP est ajoutée à Block List.

1.2. Exploitation de la faille :

Maintenant l'attacker va essayer d'exploiter toutes les failles et les informations qu'il obtient pour pirater les serveurs.

Premièrement : Brute-Force Attaque pour déchiffrer le nom d'utilisateur et le mot de passe du service Secur Shell.

Nmap -script ssh-brut-script-args : il effectue l'attaque «Brute-Force » pour qu'il obtient le nom d'utilisateur et le mot de passe en utilisant le fichier username et le fichier password.

```
hacker@hacker:~/Bureau$ ls

 passwords.txt usernames.txt

 hacker@ahacker:~/Bureau$

 hacker@hackerr:~/Bureau$ nmap --script ssh-brute --script-args
userdb=./usernames.txt,passdb=./passwords.txt 192.168.1.5 -p 22

 Starting Nmap 7.80 ( https://nmap.org ) at 2022-06-16 17:57 CEST

 NSE: [ssh-brute] Trying username/password pair: admin:admin

 NSE: [ssh-brute] Trying username/password pair: administrator:administrator

NSE: [ssh-brute] Trying username/password pair: toor:toor

 NSE: [ssh-brute] Trying username/password pair: root:root

 NSE: [ssh-brute] Trying username/password pair: admin:password

NSE: [ssh-brute] Trying username/password pair: administrator:password

 NSE: [ssh-brute] Trying username/password pair: admin:abc1234

 NSE: [ssh-brute] Trying username/password pair: toor:abc1234

 NSE: [ssh-brute] Trying username/password pair: administrator:passw@rd

 NSE: [ssh-brute] Trying username/password pair: toor:whatever

 NSE: [ssh-brute] Trying username/password pair: admin:football

NSE: [ssh-brute] Trying username/password pair: administrator:football

 NSE: [ssh-brute] Trying username/password pair: toor:football

 NSE: [ssh-brute] Trying username/password pair: administrator:root

 NSE: [ssh-brute] Trying username/password pair: toor:firewall

 NSE: [ssh-brute] Trying username/password pair: admin:Dwayne

 NSE: [ssh-brute] Trying username/password pair: admin:redneck

 NSE: [ssh-brute] Trying username/password pair: administrator:redneck

 NSE: [ssh-brute] Trying username/password pair: toor:Ubuntu

 NSE: [ssh-brute] Trying username/password pair: toor:Emilia

 NSE: [ssh-brute] Trying username/password pair: passw@rd:passw@rd
```

```
Nmap scan report for 192.168.1.5
 Host is up (0.038s latency).
PORT STATE SERVICE
22/tcp open ssh
| ssh-brute:
| Accounts:
|
root:root - Valid credentials          <== le mot de passe et le nom d'utilisateur
|_ Statistics: Performed 6823 guesses in 901 seconds, average tps: 7.4
Nmap done: 1 IP address (1 host up) scanned in 902.43 seconds
hacker@hacker:~/Bureau$
```

L'attaquant a pu obtenir le nom d'utilisateur et le mot de passe du serveur honeypot complètement isolé du vrai serveur, c'est-à-dire qu'il s'agit d'un faux système, il n'y a aucun danger sur le réseau car l'attaquant ne bénéficiera de rien, il ne fera que drainer son énergie, des ressources et du temps dans un lieu imaginaire qui ne contient pas d'information importantes, tout est faux pour l'ombrager et lui extraire plus d'information.

Connexion au serveur ssh-honeypot :

```
hacker@hacker:~/Bureau$ ssh root@192.168.1.5
root@192.168.1.5's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@server:/#
```

```
root@server:~#
root@server:~# cd..
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
Sbin selinux srv sys tmp usr var
vmlinuz
root@server:/# cd home/
root@server:/home# ls
Richard
root@server:/home# cd richard/
root@server:/home/richard# ls
root@server:/home/richard#
root@server:/home/richard# cd..
root@server:/home# cd..
root@server:/# cd root/
root@server:~# ls
root@server:~#
root@server:~# whoami
root
root@server:~#
```

Ce sont tous de faux fichiers qui font croire à l'attaquant qu'il est connecté à un vrai serveur, ou il peut créer des fichiers ou supprimer des fichiers sur le serveur comme l'illustration suivante :

- Ajoutez un fichier nommé vvs.txt à l'aide de la commande suivante :

```
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
Sbin selinux srv sys tmp usr var
Vmlinuz
root@server:/
root@server:/# touch vvs.txt
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
Sbin selinux srv sys tmp usr vvs.txt
var Vmlinuz
root@server:/#
```

- Supprimez le fichier vvs.txt à l'aide de la commande suivante :

Rm vvs.txt

```
root@server:/# rm vvs.txt
root@server:/
root@server:/# ls
bin boot dev etc home initrd.img lib
lost+found media mnt opt proc root run
Sbin selinux srv sys tmp usr var
Vmlinuz
root@server:/# exit
Connection to 192.168.1.5 closed.
hacker@hacker:~/Bureau
```

1.3. Visualisation de données récoltées par le honeypot :

Après le processus de piratage, nous entrons dans le MHN-Server pour voir les résultats.

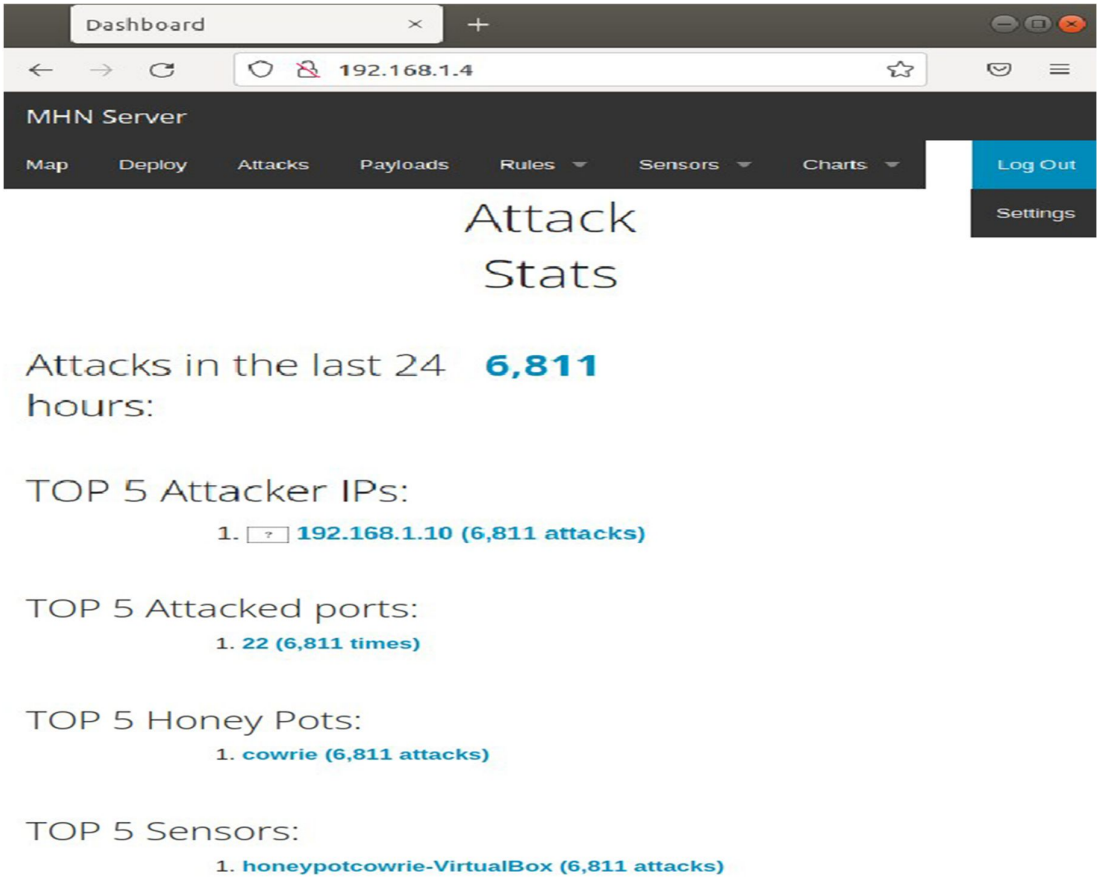

Figure 4.01: Le résultat affiché dans le tableau de bord.

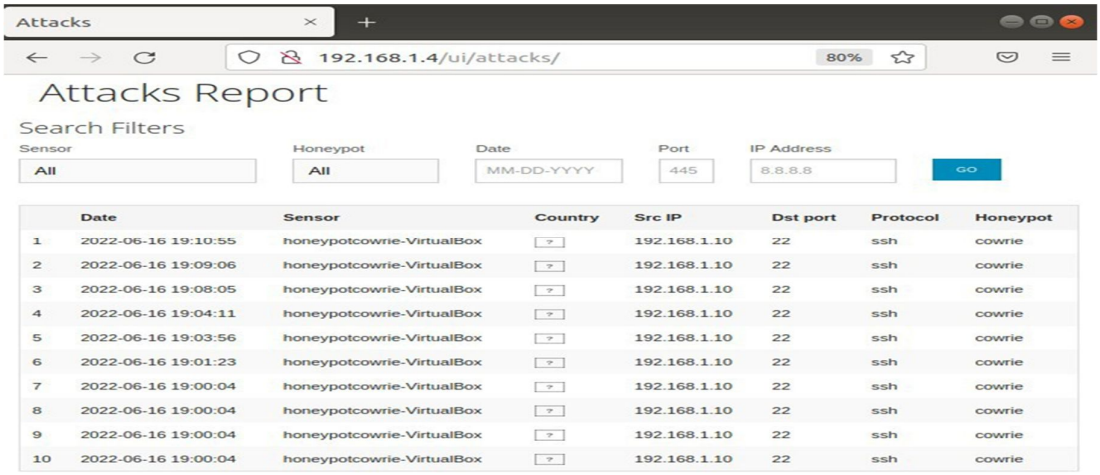

Figure 4.02: Montre des informations sur l'attaque.

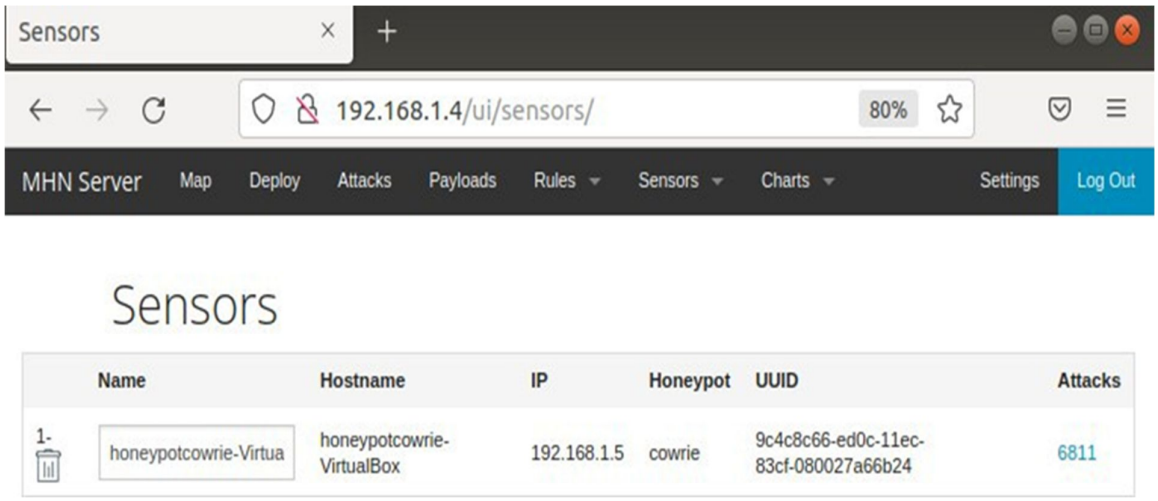

Figure 4.03: Montre le nombre d'attaque dans le capteur.

Dans le menu en haut, nous entrons dans les graphiques « charts »

Figure 4.04: Pourcentage de tentatives de connexion « top utilisateurs ».

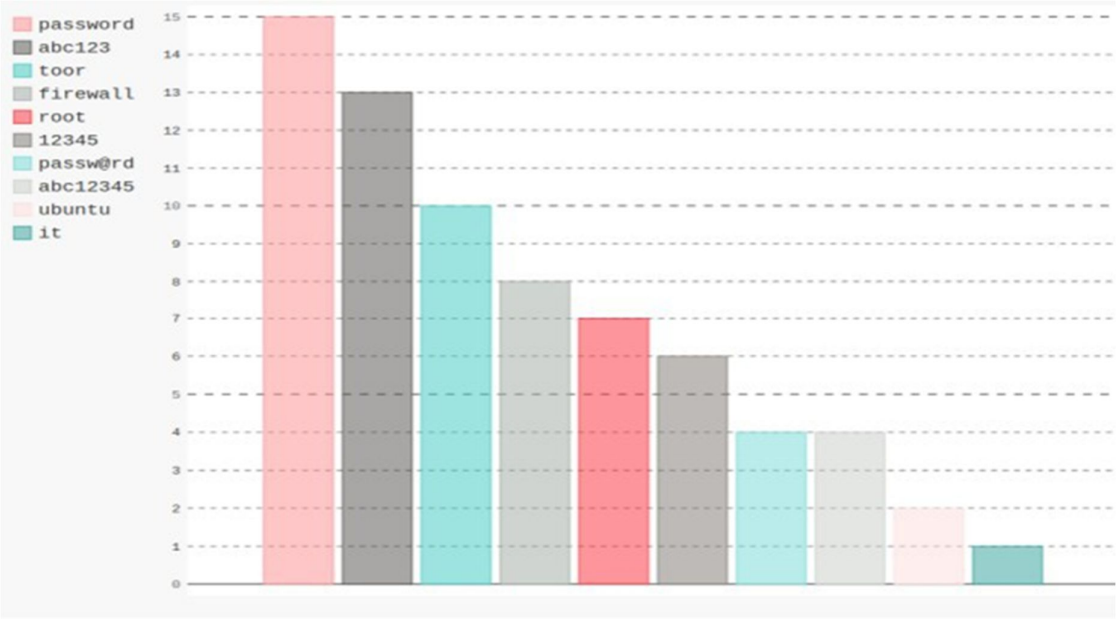

Figure 4.05: Pourcentage de tentatives de connexion « top mot de passe ».

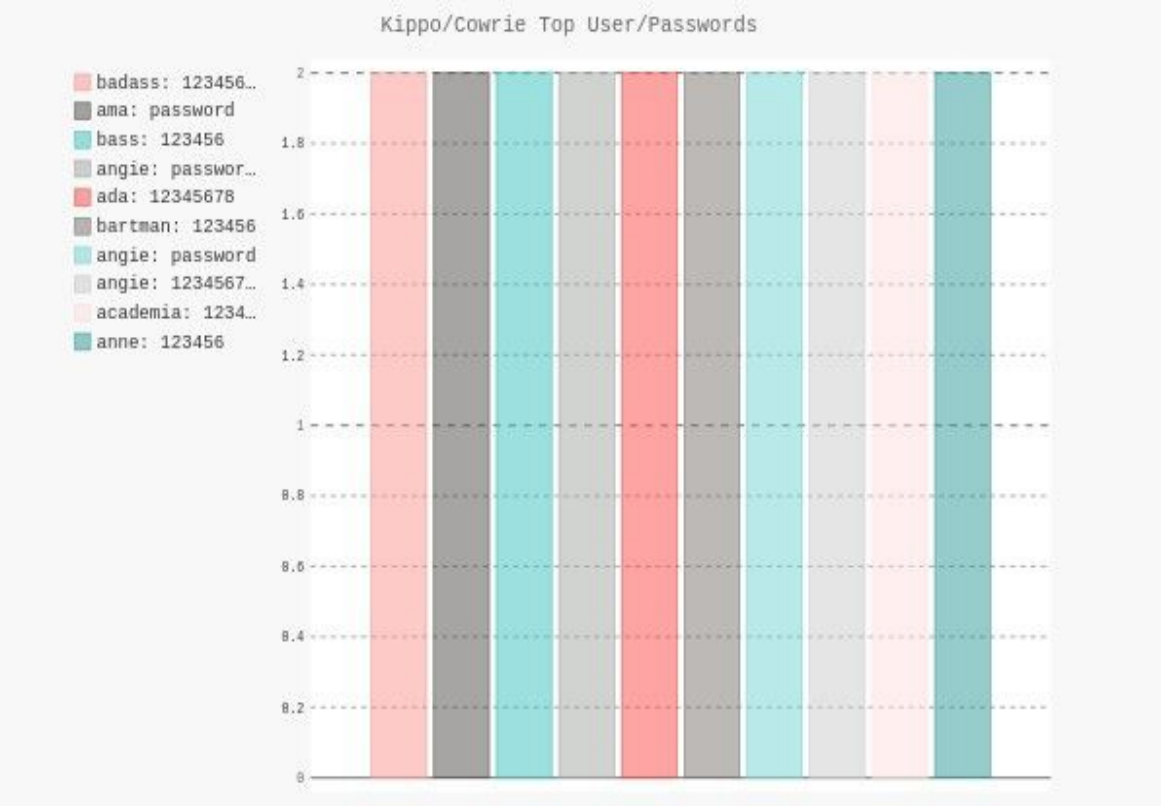

Figure 4.06: Pourcentage de tentatives de connexion « top utilisateurs mot de passe ».

Figure 4.07: Pourcentage de tentatives de connexion « top attaques ».

En cas d'attaque par plus d'un attaquant, les résultats sont les suivants :

Figure 4.08: L'affichage des résultats en cas de plusieurs attaquants.

A l'intérieur du MHN-server, il y a un outil spécial appelé Honey-map qui est une carte utilisée pour montrer l'origine des attaques qui se produisent en temps réel en fonction de l'adresse IP.

Nous entrons dans la honeymap en tapant l'adresse IP plus /3000.

Figure 4.09: Localisation de l'attaque.

2. Bilan de la simulation :

Ce bilan se portera sur les Honeypot en général mais pas seulement sur le Honeypot cowrie. Certes, il existe plusieurs types de Honeypot mais le fonctionnement général reste plus ou moins le même, c'est-à-dire, répondre de manière à satisfaire l'attaquant tout en collectant des données sur l'attaque.

Cependant, la mauvaise administration du Honeypot peut affecter l'intégrité des données collectées par le Honeypot. Les niveaux d'interaction en honeypot offrent des avantages, mais peuvent aussi parfois provoquer quelques problèmes.

3. Conclusion :

Pour conclure, ce chapitre donne un aperçu des opérations de honeypots. Dans un cas pratique, également une évaluation générale d'un honeypots.

Conclusion générale

En conclusion, honeypot s'avère être une solution adéquate pour les menaces informatiques visant les réseaux. Mais il faut tout de même prendre en compte que son implantation nécessite un investissement supplémentaire. Il est aussi nécessaire de considérer que le honeypot seul, n'est pas l'élément qui empêchera les attaques contre le réseau. Il doit être couplé avec d'autres éléments tels que les IDS/IPS et les pare-feu. Le honeypot dévoilera les intentions mais aussi les techniques utilisées par les pirates. Il va donc faire en sorte de distraire le pirate afin de l'étudier. Il est d'une importance primordiale pour une entreprise de disposer de système de protection réseau performant. La protection des données est un enjeu qui n'est pas à mettre à la légère. La catastrophe provoquée par une attaque dans un environnement de production pourrait être considérable, surtout dans les cas où le hacker cherche à corrompre ou voler des données sensibles. Le secteur honeypot réunit une communauté qui travaille sans cesse à améliorer les capacités des produits développés, notamment à travers l'organisation « The Honeynet Project ». Chaque année, le programme « Google Summer of Code » ou « GSoC » rassemble aussi des projets d'organismes dont les projets honeypot. Tout cela démontre que le honeypot est une solution de sécurité en cours de développement et sera pour le bon temps une alternative adoptée par les sociétés pour contrecarrer les tentatives d'attaques informatiques. Enfin, ce mémoire a permis d'affirmer que les technologies honeypot sont de bonnes solutions pour la sécurité des réseaux, avec évidemment leurs propres spécificités. Le honeypot est une technologie relativement jeune et encore en développement, d'où son utilisation qui est encore moins commune mais dans quelques années, il se trouve qu'il sera un des composants essentiels dans la sécurisation des systèmes informatiques. Cela réduirait considérablement le nombre d'attaques visant les réseaux et améliorera la protection des données sensibles.

Bibliographie

[1] Gunadiz, S. (2011). Algorithme d'intelligence artificielle pour la classification d'attaques réseaux à partir de données TCP (Thèse de magistère). Université M'Hamed Bougara, Boumerdes.

[2] Yende, R. (2018). SUPPORT DE COURS DE SÉCURITÉ INFORMATIQUE ET CRYPTO.

[3] MUSSET, J. (2009). *Sécurité informatique, Ethical Hacking: Apprendre l'attaque pour mieux se défendre*. ENI.

[4] Cole, E., Krutz, R., & Conley, J. (2005). *Network security bible. Wiley Publishing*, INC.

[5] Farhaoui, Y. (2012). Evaluation des systèmes de détection et de prévention des intrusions et la conception d'un BIDS (Thèse de doctorat). Université IBN ZOHR, Agadir.

[6] Poinsot, L. (2019). Introduction à la sécurité informatique. *Support de cours, Université Paris*, *13*.

[7] Burgermeister, D., & Krier, J. (2006). Les systèmes de détection d'intrusions. [En ligne]. Disponible à: <http://dbprog.developpez.com. > [Consulter le 15 mai 2022]

[8] Rutkowska, J. (2006). Introducing stealth malware taxonomy. *COSEINC Advanced Malware Labs*, 1-9.

[9] Carpentier, J. F. (2009). *La sécurité informatique dans la petite entreprise: état de l'art et bonnes pratiques*. Editions ENI.

[10] Bloch, L., Wolfhugel, C., Queinnec, C., Schauer, H., & Makarévitch, N. (2013). *Sécurité informatique Principes et méthodes à l'usage des DSI, RSSI et administrateurs*. Editions Eyrolles.

[11] Llorens, C., Levier, L., & Valois, D. (2006). *Tableaux de bord de la Sécurité réseau* (2e édition). Editions Eyrolles.

[12] Pillou, J. F., & Bay, J. P. (2013). *Tout sur la sécurité informatique* (3eme édition). Dunod.

[13] Detoisien, E. (2005). Les attaques externes. IEEE Transactions on software engeneering. [En ligne]. Disponible à: < http://linuxfocus.org. > [Consulter le 15 mai 2022]

[14] Cheikh, M. (2009). Un mecanisme de détection d'intrusion Adaptative basé sur les agents. Ecole Doctorale d'Informatique de l'Est, Pôle Constantine.

[15] Le grand livre de la sécurité informatique. *SecuriteInfo*, Editions du 6 novembre 2006.

[16] Viardin, A., & Latu, P. (2003). Un petit guide pour la sécurité. [En ligne]. Disponible à: <https://www. inetdoc. net/guides/tutoriel-secu. > [consulter le 16 mai 2022]

[17] Pillou, J. F., & Bay, J. P. (2009). *Tout sur la sécurité informatique* (2eme édition). Dunod.

[18] Lussan, P. (2022) Les 10 types de cyberattaques les plus courants. Blog de Netwrix. [En ligne]. Disponible à: < http:// blog.netwrix.fr. > [Consulter le 17 mai 2022]

[19] Chikh, A., & Djennane.A. (2012). Sécurité d'une application Web à l'aide d'un système de détection d'intrusions comportementale. Université Abou Bakr Belkaid, Tlemcen.

[20] Jabou, C., Schillings, M., & Hantach, A. (2009). TER Detection d'anomalie sur les réseaux.

[21] ENONGA, S. (s. d.). Le paradigme de la relation banque - clients dans les services bancaires sur internet. [En ligne]. Disponible à: < http://www.memoireonline.com. > [Consulter le 22 mai 2022]

[22] Tay, T. V. (2005). Les systemes de détection d'intrusion et les systèmes d'empêchement des intrusions. Institut de la Francophonie pour l'Informatique: Montréal.

[23] Spitzner, L. (2002). *Honeypots: tracking hackers* (Vol. 1). Reading: Addison-Wesley.

[24] Gerrit Göbel Jan, Dewald Andreas. (2011). *Client-Honeypots: Exploring Malicious Websites*. München: Oldenbourg.

[25] The Honeynet Project, Papers-Know Your Enemy: The Social Dynamics of Hacking. [En ligne]. Disponible à: < http://www.honeynet.org/paper. > [Consulter le 26 mai 2022].

[26] Lakhani, A. D. (2003). Deception techniques using Honeypots. *MSc, University of London, UK*.

[27] Spitzner, L. (2003). Honeypots: Definitions and value of honeypots.

[28] W. Ren and H. Jin, "Honeynet based distributed adaptive network forensics and active real time investigation," in Proceedings of the 2005 ACM symposium on Applied computing, Saanta Fe, New Mexico, 2005, pp. 302–303.

[29] Joshi, R. C., & Sardana, A. (Eds.). (2011). *Honeypots: a new paradigm to information security*. CRC Press.

[30] Provos, N., & Holz, T. (2007). *Virtual honeypots: from botnet tracking to intrusion detection*. Pearson Education.

[31] Mohammed, M., & Rehman, H. U. (2015). *Honeypots and Routers: Collecting internet attacks*. CRC Press.

[32] Artail, H., Safa, H., Sraj, M., Kuwatly, I., & Al-Masri, Z. (2006). A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks. *computers & security*, *25*(4), 274-288.

[33] Boulaiche, A. (2008). Technologies Honeypots (Thèse de magistère). *Université Abderrahmene Mira, Béjaia*.

[34] Kevat, S. M. (2017). Review on Honeypot Security. *International Research Journal of Engineering and Technology (IRJET)*, *4*(06), 1200-1203.

[35] Fernandes, D., & Sarr, P. (2010). La protection des réseaux contre les attaques Dos. *Université Paris Descarte*.

[36] Peter, E., & Schiller, T. (2011). A practical guide to honeypots. *Washington Univerity*.

[37]Yek, S., & Valli, C. (2003). If you go down to the Internet today–Deceptive Honeypots. *Journal of Information Warfare*, *2*(3), 101-108.

[38] Matin, I. M. M., & Rahardjo, B. (2020, October). A Framework for Collecting and Analysis PE Malware Using Modern Honey Network (MHN). In *2020 8th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE.

[39] Michel Oosterhof,cowrie documentation Release 19.10.0. [En ligne]. Disponible à: <https://github.com/cowrie/cowrie/> [Consulter le 10 juin 2022]