République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU **G**ENIE **E**LECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Domaine : Mathématiques et Informatique
Filière : Informatique

Spécialité: **Réseaux, Mobilités et Systèmes Embarqués.**

Présenté par
Haddad Amélia

Thème

Sécurisation d'un environnement Cloud grâce aux Honeypots.

Soutenu publiquement le 30/09/ 2017 devant le jury composé de :

Président :	**Mme**	**Rachida**	**Aoudjit**
Encadreur :	**Mr**	**Mehammed**	**Daoui**
Examinateur :	**Mme**	**Malika**	**Belkadi**
Examinateur :	**Mme**	**Karima**	**Oukfif**

2016/2017

Remerciements

C’est avec un grand plaisir que je réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui ont bien voulu apporter l’assistance nécessaire au bon déroulement de ce travail.

J’aimerais, tout d’abord, exprimer toute ma gratitude à mon encadreur Mr Daoui pour la qualité de son encadrement, de son aide pour mener à bien mes recherches et de sa disponibilité.

J’adresse mes sincères remerciements à tous les membres du jury, Mme Aoujit autant que présidente, Mme Belkadi et Mme Oukfif autant qu’examinatrices, qui m’ont fait l’honneur d’accepter de prendre part à ce jury et de porter un œil critique à mon travail.

Je tiens aussi à remercier Mlle Lynda Boukela d’avoir proposé ce thème, et d’avoir suivi l’évolution de mon travail pas à pas toujours en faisant des remarques très constructives.

Je souhaite aussi remercier les doctorantes du LARI de m’avoir accueillie au sein de leur équipe et de m’avoir permis de travailler dans d’excellentes conditions.

Merci à ma famille et à mes amis de m’avoir encouragée dans les moments de doutes et d’avoir cru en moi.

Dédicace

J'aimerais dédier ce travail en premier lieu à mes parents qui m'ont toujours soutenue et poussée à donner le meilleur de moi-même ainsi qu'à :

-Mes sœurs Nassima et Manel, leurs maris Sid-Ali et Karim, à ma nièce Anaïs et mon neveu Elias.

-Toute ma famille particulièrement : ma grand-mère, mes tantes, mes cousines Melissa, Yasmine, Lynda et à tonton Moh.

-Mes amis et à toute personne m'ayant aidée et soutenue.

Amélia.

Sommaire

Acronymes: 8
Introduction générale : 10
Chapitre I : Le Cloud Computing. 12
Introduction: 13
I. Définition : 13
II. Historique : 13
III. Les principales caractéristiques du cloud computing : 15
IV. Modèles du Cloud Computing (classification SPI): 16
V. Les types de déploiement du cloud : 20
VI. Complémentarité entre Cloud et virtualisation: 21
VII. Principaux chalenges du Cloud : 21
1. Les problèmes de sécurité: 21
2. Standardisation et interopérabilité : 24
3. Qualité de service: 26
4. Transfert de données lent: 27
VIII. Les principaux systèmes IaaS : 27
1. OpenStack : 27
2. Cloudstack : 28
3. Eucalyptus : 28
4. OpenNebula : 30
IX. Comparaison entre différentes solutions IaaS : 32
Conclusion: 32
Chapitre II : La Sécurité informatique,Honeypots et Honeynets. 33
Introduction: 34
I. Sécurité d'un réseau 34
II. Évaluation de la sécurité d'un réseau: 34
III. Raisons qui poussent à la sécurisation des réseaux : 35
1. Les enjeux : 35
2. Les vulnérabilités : 36
3. Les Menaces : 36
4. Les Risques : 37
IV. Pirates et attaques informatique : 37
1. Haker 37
2. Les logiciels malveillants : 37
3. Motivation des attaques : 39
V. Mécanismes de sécurité : 40
1. Cryptage : 40
2. Pare-feu : 40
3. Antivirus : 40
4. VPN : 41
5. IDS : 41
6. IPS : 41

VI. Définition d'un Honeypot : 42
VII. Historique des Honeypots: 42
VIII. Caractéristiques d'un honeypot : 43
IX. Les types de Honeypots : 43
1. Selon le Niveau d'interaction : 43
2. Selon le type de ressources attaquées : 46
3. Selon l'environnement de mise en place: 47
4. Selon les données capturées : 47
5. Selon le contenu: 48
6. Selon la distribution : 48
7. Selon l'interface de communication : 48
X. Avantages et inconvénients d'un Honeypot : 49
XI. Avantages face à un simple IDS : 50
XII. Les Honeynets 50
XIII. Architecture d'un honeynet : 50
XIV. Considérations et conditions de déploiement : 51
1. Contrôle des données : 51
2. Capture de données : 52
3. Sauvegarde de données : 52
4. L'analyse des données : 53
XV. Les Honeypots dans le Cloud Computing : 53
Conclusion: 54
Chapitre III : Présentation des outils utilisés. 55
Introduction : 56
I. Openstack : 56
1. Definition : 56
2. Les composants clés d'OpenStack : 57
3. Architecture minimale : 57
a. Controller (contrôleur) : 57
b. Compute (Calcule) : 58
II. DevStack : 58
1. Définition : 58
2. Installation d'OpenStack via DevStack: 58
III. Dionaea : 62
1. Protocoles à partir desquels Dionaea piège les Malwares : 62
2. Fonctionnement : 63
3. Installation : 64
IV. P0f : 66
1. Capacités de p0f : 66
2. Fonctionnement de p0f: 67
3. Installation et lancement de p0f: 69
V. Honeywall : 69
Mise en place : 70

VI. Nmap : 70
1. Caractéristiques ne nmap : 71
2. Fonctionnement : 72
3. Ouvrir Nmap : 73
VII. Netcat : 73
1. Utilisation de netcat: 74
2. Ouvrir Netcat : 75
Conclusion: 76
Chapitre IV : Tests et résultats. 77
Introduction: 78
I. Topologie de notre réseau : 78
II. Scan de ports avec Nmap : 79
1. Résultat de p0f : 80
2. Fichier Log de Dionaea : 81
III. Attaque netcat : 82
1. Résultat p0f : 83
2. Fichier log de Dionaea : 83
Conclusion: 84
Conclusion Générale: 85
Annexe : Création d'une instance OpenStack : 86
Références : 88

Liste des figures :

Chapitre I :
Figure 1: Le modèle pyramidal SAAS, PAAS, IAAS [7] 16
Figure 2: Architecture du gestionnaire Eucalyptus. [13] 29
Figure 3: Architecture du projet OpenNebula. [15] 31
Chapitre II:
Figure 4: Honeypot à faible interaction. [38] 44
Figure 5: Honeypot à forte interaction. [38] 45
Figure 6: Honeypot à moyenne interaction. [38] 45
Figure 7: Architecture d'un honeynet. [44] 51
Chapitre III:
Figure 8: Ressources allouées pour la création de la machine virtuelle pour OpenStack. 59
Figure 9: Screenshot de note VM à la fin de l'installation d'OpenStack via DevStack. 60
Figure 10: Champs d'identification dans le dashboard. 61
Figure 11: Schéma de fonctionnement de Dionaea. 63
Figure 12: Lancement de Dionaea. 66
Figure 13: Entête d'une trame TCP/IP. [55] 67
Figure 14: Champ service sur 8bits. [56] 68
Figure 15: Détection de ports ouverts/fermés avec nmap. 72
Figure 16: Ouverture de nmap sur une machine Kali Linux. 73
Figure 17: Exemple de shell binding avec Netcat. [61] 74
Figure 18: Ouverture de Netcat sur une machine Kali Linux. 76
Chapitre IV:
Figure 19: Topologie de notre réseau. 78
Figure 20: Scan nmap du Honeypot. 79
Figure 21: Résultats de p0f suite au scan de ports. 80
Figure 22: Fichier log de Dionaea après le scan de ports. 81
Figure 23: Connexion à distance sur le port 80 via netcat. 82

Figure 24: Résultat p0f après la connexion sur le port 80. 83
Figure 25: Résultat du fichier log de Dionaea après la connexion sur le port 80. 83
Annexe:
Figure 26: Connexion avec l'utilisateur demo. 86
Figure 27: Créer une image. 86
Figure 28: Créer une instance. 87
Figure 29: Ajout de script de personnalisation. 87

Acronymes:

API: Application Programming Interface

ARPA: Advanced Research Projects Agency

ASP: Application Service Provider

AT&T Bell : American Telephone & Telegraph Company Bell

BaaS: Backend as a Service

CDMI: Cloud Data Management Interface

CERT: Computer Emergency Readiness ou Response Team

CIMI: Cloud Infrastructure Management Interface

CISCO: Commercial & Industrial Security Corporation

CMP: Internet Control Message Protocol

CRM: Client Relation Manager

CSA: Cloud Security Alliance

DaaS: Desktop as a Service ou Data-as-a-Service

DF: Don't Fragment

DNS: Domain Name Service

DRaaS: DisasterRecovery-as-a-Service

FTP: File Transfert Protocol

GUI: Graphical User Interface

HTTP: Hypertext Transfer Protocol

IaaS: Infrastructure as a Service

IBM: International Business Machines

ICMP: Internet Control Message Protocol

IDS: Intrusion Detection System

IP: Internet Protocol

IPS: Intrusion Prevention System

IT: Information Technology

KVM: Kernel Virtual Machine

MF: More Fragments

MIT: Massachusetts Institute of Technology

MSSQL: Microsoft SQL Server

NAT: Network Address Transfer

NFQ: National Framework of Qualification

Nginx: Engine X

NIST: National Institute of Standards and Technology

Nmap: Network Mapper

NTP: Network Time Protocol

OCCI: Open Cloud Computing Interface

OS: Operating System

OSI: Open System Interconnection

OVF: Open Virtualization Format

P0f: Passive Operating system Fingerprinting

PaaS: Platform as a Service

PPA: Personal Package Archives

QoS: Quality of Service

RAM: Random Access Memory

SaaS: Software as a Service

SCADA: Supervisory Control and data Acquisition

SGNET: Secure Integrated Global Network

SI : Système Informatique

SIP : Session Initial Protocol

SLA: Service Level Agreement

SMTP: Simple Mail Transfer Protocol

SPI: Service Provider Interface

SSH: Secure Shell

TCP: Transmission Control Protocol

TFTP: Trivial File Transfert Protocol

TOS: Type Of Service

TOSCA: Topology and Orchestration Specification for Cloud Applications

TTL: Time To Live

UDP: User Data Protocol

USB: Universal Serial Bus

VM: Virtual Machine

VoIP: Voice over IP

VPN: Virtual private Network

Introduction générale :

L'informatique en nuage est de plus en plus utilisée, à la fois par les particuliers et les entreprises. En effet, de nombreuses entreprises (Google, Dropbox, Amazon, etc.) fournissent des services reposant sur l'informatique en nuage ou permettent d'utiliser une infrastructure en nuage pour déployer des services. De plus, l'importance économique de l'informatique en nuage continue d'augmenter comme montré dans une étude publiée en 2014 **[62]**. Selon cette dernière en 2013 le marché de l'informatique en nuage représentait 58 milliards de dollars. L'étude prévoit que ce marché devrait atteindre 191 milliards de dollars en 2020. Cette forte augmentation témoigne donc de l'importance croissante de l'informatique en nuage.

Cependant, selon une enquête du CSA (Cloud Security Alliance), **[63]** l'un des freins majeurs à l'adoption de l'informatique en nuage concerne les problèmes de sécurité des données, puisque 73% des participants interrogés s'en inquiètent. Les fournisseurs de services en nuage doivent donc apporter une réponse aux questions de leurs utilisateurs concernant la sécurité : en effet, les deux principaux critères sur lesquels les entreprises évaluent les fournisseurs de services **[64]** sont la sécurité des données (82%) et la vie privée (81%), alors que le critère de coût n'arrive qu'en troisième position (78%). La sécurité est donc un élément crucial pour les clients et par conséquent pour les fournisseurs de services. Ainsi, malgré l'importance croissante de l'informatique en nuage, les problèmes de sécurité persistent et doivent être adressés.

L'objectif de ce travail est de fournir une solution de sécurité pour les environnements de cloud computing à travers les techniques liées aux Honynets (réseau de Honypots). Pour cela, nous avons décidé dans un premier temps de créer un environnement de Cloud Computing et dans un second temps de doter cet environnement d'un système de sécurité basé sur les Honeypots pour garder un œil sur ce qui se fait en matière de piratage et de créer de meilleurs systèmes de défenses.

Pour mener à bien notre travail, nous avons opté pour une démarche qui s'étale sur quatre (04) chapitres :

- Le premier a pour objectifs d'introduire la notion de Cloud Computing et de nous aider à choisir la meilleure solution pour la suite de notre travail.

- Le second présentera les différents types d'attaques informatiques et des motivations des attaquants ainsi que des objectifs de la sécurité informatique notamment ceux des Honeypots et Honeynets.
- Le troisième sera une présentation des différents outils utilisés, aussi bien ceux pour mettre en place le Cloud et le Honeynet que ceux pour simuler des attaques informatiques.
- Le quatrième et dernier chapitre sera un peu le résultat de notre travail car il présentera les tests effectués sur notre environnement ainsi que leurs résultats.

Nous clôturerons ce mémoire par une conclusion générale sur le travail accompli ainsi que ses perspectives.

Chapitre I : Le Cloud Computing.

Introduction:

Bien qu'il soit sur le marché depuis un certain temps, le Cloud Computing est encore une source de confusion pour beaucoup. Bon nombre d'entre nous utilisent déjà un Cloud sans même le savoir. Il est difficile de trouver une définition du Cloud Computing acceptée par tous. Ce chapitre présente l'état de l'art, formule et élabore une définition unifiée des objectifs du Cloud Computing. Il donne également une vision globale de ce qu'il apporte réellement.

I. Définition :

Parmi les définitions les plus utilisées :

NIST (National Institute of Standards and Technology): **[65]** « Le cloud computing est un modèle qui permet un accès réseau simple et pratique, à la demande, à un pool partagé de ressources informatiques configurables (telles que réseaux, serveurs, stockage, applications et services). Ces ressources peuvent être provisionnées rapidement et distribuées avec un minimum de gestion ou d'interaction avec le fournisseur de services. »

CISCO **[66]:** «Le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité des ressources».

Le Cloud Computing est un concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur. Il consiste à proposer des services informatiques sous forme de service à la demande, accessible de n'importe où, n'importe quand et par n'importe qui.

II. Historique :

-En 1961 la notion de ce service a été proposée pour la première fois lors d'une conférence au MIT (Massachusetts Institute of Technology) par John McCarthy.

- En 1962 : lancement de la recherche par l'ARPA sur la création d'un réseau global d'ordinateurs fondé sur la commutation de paquets.

- En 1969 : naissance du Netwok Working Group qui connecte des ordinateurs pour la première fois.

- En 1971 : envoi du premier courriel par Ray Romlinson.

- **En 1990**, un nouveau type d'entreprise est apparue : les fournisseurs d'applications en ligne (Application Service Provider, ASP) ancêtres du Software as a Service (SaaS). Les ASP achètent du matériel informatique et développent des applications métiers Internet, puis elles les exécutent continuellement pour leurs clients moyennant le paiement d'un abonnement mensuel.

- **En 1991**, 1 000 000 ordinateurs connectés et 36 000 000 en 1996. Dès lors les universitaires se penchent sur la façon de connecter ces machines pour créer un stockage massif et partagé. C'est alors que l'idée de« réseau » (grid) a commencé à prendre forme. Le terme « réseau » est interprété comme un synonyme de « nuage », car les deux sont faits avec l'ajout de nombreux ordinateurs connectés.

-**En 1999** salesforce.com fut le premier hébergeur de Cloud, suivi en 2002 par Amazon qui proposa un ensemble d'hébergement d'application et de stockage. Amazon développa ses services en 2005 (Amazon Web Services) et en 2006 (Elastic Compute Cloud ou EC2).
-**En 2007**, Google, IBM et des universités lancèrent un projet de recherche sur le Cloud qui permit de lui faire gagner en popularité et en consistance.
-**En 2009** c'est là que la réelle explosion du Cloud survint avec l'arrivée sur le marché de sociétés comme Google (Google App Engine), Microsoft (Microsoft Azure), IBM (IBM Smart Business Service), Sun (Sun Cloud) et Canonical Ltd (Ubuntu Enter- prise Cloud).

Depuis, le cloud n'a cessé de croître : en 2013, les dépenses mondiales en services cloud étaient estimées à 47 milliards de dollars. Nous sommes en 2017, ce chiffre a plus que doubler puisqu'il a atteint 122.5 milliards de dollars. Ces chiffres ne feront que croitre à mesure que les entreprises investissent dans des services cloud pour créer de nouvelles offres compétitives. **[1][24]**

III. Les principales caractéristiques du cloud computing :

- **Le libre-service à la demande** : La notion de libre-service à la demande est primordiale pour les utilisateurs du cloud. Le libre-service à la demande permet à l'utilisateur d'être en mesure d'allouer, mais également de libérer des ressources distantes en temps réel en fonction des besoins, et sans nécessiter d'intervention humaine.

- **L'accès universel** : L'ensemble des ressources doit être accessible et à disposition de l'utilisateur universellement et simplement à travers le réseau, quels que soient les clients utilisés (serveur, PC, client mobile, etc.). Ceci implique trois prédispositions :

- Accès utilisateur : afin de garantir que l'accès à une solution de cloud soit perçu comme stable et fiable, il est nécessaire que celle-ci dispose d'un accès haut débit.
- Accès applicatif : en vue de s'assurer que la solution soit en mesure de mettre à disposition de l'utilisateur les autres fonctionnalités du cloud, l'accès au réseau doit être disponible au sein même de la solution.
- Accès opérateur : dans le but de pouvoir maintenir et administrer un système cloud correctement, le fournisseur de service cloud doit pouvoir accéder à celui-ci via le réseau. Ainsi, s'assurer qu'un accès total au réseau est disponible dans tous les aspects d'une solution de cloud est essentiel pour offrir l'ensemble des autres caractéristiques.

- **Mutualisation des ressources :** De multiples ressources matérielles son regroupées et partagée de la par du fournisseur entre les différents utilisateurs du service. Ce partage rend l'emplacement exact des données des utilisateurs impossible à déterminer, ce qui peut poser un véritable problème aux entreprises qui sont soumises à de nombreuses contraintes réglementaires concernant la localisation et le contrôle des données. En contrepartie, les économies d'échelle réalisées permettent de réduire les coûts du fournisseur et donc les dépenses des utilisateurs.
- **Adaptabilité rapide :** L'une des caractéristiques du cloud computing est l'élasticité des ressources. Cette caractéristique permet aux utilisateurs d'allouer rapidement de nouvelles ressources de manière à être en mesure de répondre à une montée ou à une descente en charge soudaine. Il n'est jamais évident de prévoir les ressources qui seront nécessaires à la mise en place d'un service informatique quelconque, en particulier lorsque ce besoin est en constante évolution. Le cloud computing offre ainsi un moyen de fournir les ressources informatiques nécessaires à une évolution ou à un pic d'utilisation de ce service.

- **Paiement à l'usage :** Les systèmes cloud doivent être capables de s'autocontrôler et de se gérer pour permettre l'optimisation interne du système. Pour cela, ils s'appuient sur des mesures de référence obtenues grâce à divers mécanismes de supervision. Ces mesures précises permettent une juste facturation des utilisateurs ; ceux-ci ne payeront ainsi que pour les ressources qu'ils ont utilisées et seulement pour la durée qu'ils les ont utilisées. **[2]**

IV. Modèles du Cloud Computing (classification SPI):

La classification SPI propose de situer un service de Cloud Computing au sein de trois couches : Infrastructure, Platform et Software. Cet ordre correspondant à un degré d'abstraction croissant de l'infrastructure sous-jacente comme le montre la figure 1. Dans ce modèle, les services appartenant aux couches supérieures font abstraction des fonctions relevant des couches inférieures : cela correspond à une observation de la réalité où il est courant qu'un service s'appuie sur d'autres services appartenant à des couches plus basses dans la classification.

La suite de cette partie décrit les différentes couches constitutives du modèle SPI, allant des couches d'abstraction les plus hautes vers les plus basses, permettant ainsi de mettre en évidence le rôle des couches proches de l'infrastructure. **[3]**

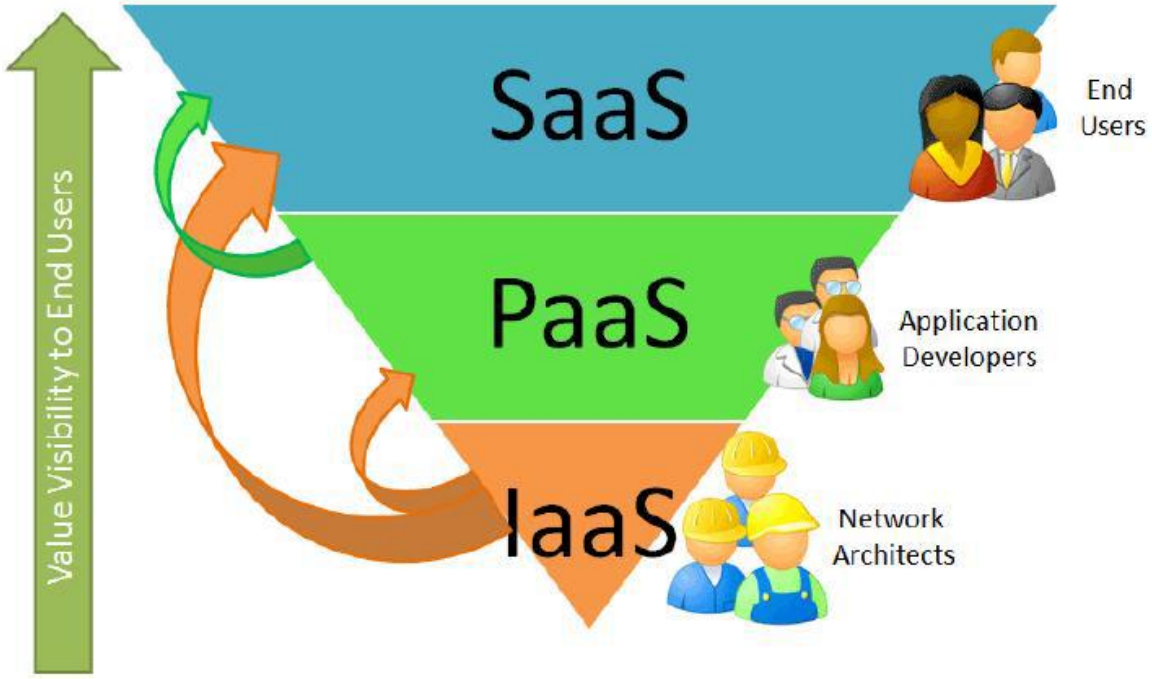

Figure 1: Le modèle pyramidal SAAS, PAAS, IAAS [7]

• Software as a Service (SaaS) :

La couche haute Software as a Service (SaaS) regroupe des services rendus accessibles aux utilisateurs finaux (application web, application mobile, bureau, . . .). Ces services sont hébergés à distance : La majeure partie de leur code source, est exécutée à distance sur les infrastructures des fournisseurs de services.

Les avantages d'un tel modèle de service sont nombreux, comme le fait que la complexité liée au déploiement d'un service ainsi que sa maintenance sont masquées à l'utilisateur final. Les exigences en matière de matériel informatique sont aussi plus basses, car une partie de l'exécution du service est délocalisée sur l'infrastructure distante appartenant au fournisseur. L'utilisateur final bénéficie d'une garantie de qualité de service sans avoir à investir dans du matériel informatique spécialisé et performant, et en cas de pannes le contrat de garantie prévoit les indemnités que le fournisseur versera à l'utilisateur.

Cependant, toutes les données des utilisateurs se trouvant sur l'infrastructure, en cas d'intrusion celles-ci se retrouvent exposées, de même qu'en cas de panne de l'infrastructure, il est fréquent que le service devienne totalement inaccessible pour les utilisateurs.

Les exemples connus de SaaS sont le service de messagerie Gmail (Google), le réseau social Facebook et l'outil de gestion de la relation client (CRM) Salesforces.

• Platform as a Service (PaaS) :

La couche intermédiaire Platform as a Service regroupe des services qui ciblent les concepteurs de services logiciels. Les fournisseurs de PaaS fournissent des plates-formes logicielles aux concepteurs de service, qui sont livrées avec des interfaces de programmation dédiées (API) pouvant accueillir et exécuter des services de Cloud Computing. En utilisant les services de la couche PaaS, le travail des concepteurs est simplifié, car une grande partie de la complexité liée à la prise en main des aspects liés à l'infrastructure est masquée derrière l'API exposée par le fournisseur PaaS. De cette manière, le processus de développement d'un service logiciel est simplifié, car un concepteur qui utilise une plateforme PaaS ne fait que fournir du code source au fournisseur PaaS, qui en retour s'occupe de le déployer sous la forme d'un service, et de le rendre accessible aux utilisateurs au moyen d'un terminal.

Bien que le modèle de service PaaS réduise la complexité lors du développement d'un service, le code source développé doit néanmoins répondre à des contraintes dictées par les choix technologiques du fournisseur PaaS, telles que le support d'un nombre limité de langages de programmation, l'obligation d'utiliser certaines bibliothèques ou certains serveurs de bases de

données. Ces contraintes sont souvent nécessaires pour garantir au fournisseur de PaaS une intégration plus facile entre les services des utilisateurs et les technologies utilisées par son infrastructure, rendant plus facile l'hébergement de ceux-ci.

Enfin, bien que les éléments techniques composant les couches inférieures (notamment la partie IaaS) ne soient pas directement accessibles à un concepteur de service utilisant une plate-forme PaaS, cette dernière est capable de faire des ajustements concernant les quantités de ressources qui lui sont allouées, afin de rester adaptées à la charge de travail du service qu'elle héberge. De même, il est courant que les fournisseurs PaaS proposent des interfaces logicielles (APIs) à leurs utilisateurs, permettant à ces derniers de modifier les paramètres associés à l'environnement d'exécution, notamment la puissance de calcul allouée à un service.

Parmi les principaux fournisseurs de ressources PaaS, figurent des acteurs comme
Amazon (Amazon Web Service), Google (App Engine), Microsoft (Azure) et Salesforce (Heroku).

• Infrastructure as a Service (IaaS) :

La couche basse Infrastructure as a Service (IaaS) regroupe les services qui fournissent des ressources tel qu'un espace de stockage et des connexions réseaux sous la forme de machines virtuelles (VMs) aux utilisateurs.

Ces machines virtuelles sont hébergées sur des serveurs physiques localisés dans des centres de données (datacenters) appartenant au fournisseur IaaS. L'utilisation des techniques de virtualisation procure notamment l'avantage de pouvoir consolider le degré d'utilisation des serveurs physiques en hébergeant plusieurs VMs sur un même serveur physique, ce qui permet d'améliorer le taux de rentabilité d'une infrastructure. De plus, les techniques de virtualisation permettent d'avoir des propriétés d'isolation empêchant les interférences entre des machines virtuelles appartenant à différents utilisateurs.

Lors de la fourniture d'une machine virtuelle, l'utilisateur a les mêmes possibilités d'utilisation que s'il avait affaire à une machine physique. Ainsi, il peut y déployer un système d'exploitation (OS) parmi ceux proposés par le fournisseur IaaS (certains fournisseurs laissent la possibilité d'installer un OS personnalisé), modifier les fichiers de configuration du serveur et y installer ses logiciels et les bibliothèques de son choix.

Il est courant que le fournisseur IaaS propose, en complément de l'approvisionnement en machines virtuelles, des services additionnels tels que la fourniture et l'hébergement d'images pour machines virtuelles, des ressources de stockage supplémentaires, des fonctions liées à la

communication inter-VMs comme les réseaux virtuels, ou des garanties supplémentaires concernant la qualité de service (QoS) fournie pour l'hébergement de VMs.

De nombreux autres acronymes en 'aaS' émergent de jour en jour, et sont associés au Cloud Computing qui s'affiche désormais comme un mode de consommation des services IT. Il devient fastidieux de les décrire tous. Nous citrons dans ce qui suit certains d'entre eux :

- **Desktop as a Service ou Data-as-a-Service (DaaS) :**

Un même acronyme pour deux significations (Data ou Desktop).

En ce qui concerne le **Desktop as a Service,** il permet de mettre à disposition un bureau partagé. Dans ce bureau, on va retrouver des logiciels installés et des espaces de stockage. L'avantage du DaaS est que vous pouvez vous connecter à ce bureau de n'importe où avec un ordinateur, un smartphone, une tablette, une télévision, etc… Et que vous retrouvez sur l'écran un bureau qui permet de faire la même chose que si vous étiez devant un ordinateur. **[4]**

- **Data-as-a-Service :**

C'est autre chose, c'est un concept qui commence à se répandre, car il simplifie le stockage des données et confie au fournisseur le soin de supporter les variations de volume de données stockées, que ce stockage soit permanent ou temporaire, et avec une solution contractuellement sécurisée. Les modèles de facturation s'appliquent à l'usage. Ils sont souvent accompagnés de clauses de volumétrie afin que le fournisseur, puisse anticiper le dimensionnement de son infrastructure. La facturation prend en compte le volume des données, parfois le type des données (par exemple, une donnée stratégique peut bénéficier d'un traitement particulier), et de l'utilisation de ces données (trafic et bande passante consommée). Il existe également des solutions au forfait**. [6]**

- **Backend as a Service (BaaS) :**

C'est une approche du cloud computing qui fournit un backend pour les applications (principalement mobiles). Ils fournissent une API et des outils permettant à différentes langages d'ordinateur de s'intégrer à leur back-end. Ils fournissent également des services supplémentaires comme le stockage, les analyses, les notifications Push, les tableaux de bord, l'intégration

sociale. En quelque sorte, c'est similaire à SaaS, mais BaaS est principalement destiné aux développeurs, où SaaS s'adresse aux utilisateurs finaux.

Parse est le BaaS le plus connu et a été acquis par Facebook en 2013. Fournit une intégration avec la plupart des langages informatiques et couvre tous les services nécessaires à une application. Un autre BaaS populaire est Firebase, acquis récemment par Google. Il est principalement destiné aux applications en temps réel et offre également un stockage. BaaS est fortement recommandé si la solution doit être développée très rapidement avec un backend stable. [**5]**

- **DisasterRecovery-as-a-Service (DRaaS) :**

Est considéré comme une nouvelle étape du cloud computing, particulièrement aux États-Unis qui ont été sensibilisés à la problématique de la continuité de l'activité des services dans le cloud à la suite des ouragans qui ont dévasté certaines régions et entrainé des interruptions de services. **[6]**

V. Les types de déploiement du cloud :

Il existe différentes typologies du Cloud :

1. **Le Cloud privé** Le Cloud Privé est la typologie de Cloud la plus répandu ; 73% des Clouds déployés dans les entreprises en 2013 sont des Clouds privés **[1].** Ce dernier peut se déployer sous deux formes distinctes :
   - Cloud privé interne : hébergé par l'entreprise elle-même, parfois partagé ou mutualisé en mode privatif avec les filiales.
   - Cloud privé externe : hébergé chez un tiers, il est entièrement dédié à l'entreprise et accessible via des réseaux sécurisés de type VPN.
2. **Le Cloud public** est accessible par Internet et géré par un prestataire externe. Il est ouvert au public ou à de grands groupes industriels. Cette infrastructure est possédée par une organisation qui vend des services Cloud.
3. **Le Cloud hybride ou mixte** associe l'utilisation, pour une même entreprise, d'un Cloud privé et d'un Cloud public. Ces infrastructures sont liées entre elles par la même technologie qui autorise la portabilité des applications et des données.
4. **Le Cloud communautaire** comme son nom l'indique, est utilisé par plusieurs organisations qui ont des besoins communs. Le Cloud communautaire peut héberger une

application métier très spécialisée, mais commune à plusieurs entités, qui décident de fédérer leurs efforts en construisant un cloud pour l'héberger et le gérer. Ce cloud est donc plutôt dédié à une communauté professionnelle spécifique incluant partenaires, sous-traitants, pour travailler de manière collaborative sur un même projet, ou cloud gouvernemental dédié aux institutions étatiques. **[1]**

VI. Complémentarité entre Cloud et virtualisation:

La virtualisation consiste à faire fonctionner un ou plusieurs systèmes d'exploitation comme un simple logiciel, sur un système d'exploitation fonctionnant sur ordinateur (ou serveur), au lieu de ne pouvoir en installer qu'un seul par machine.

Si virtualisation et Cloud Computing ne sont pas des termes similaires, ils reposent pourtant sur des fondamentaux communs, dans la mesure où aujourd'hui, la délivrance de services de Cloud Computing comporte quasi nécessairement de la virtualisation.
Parce que le Cloud consiste à délivrer des services disponibles dans un catalogue, il fonctionne en instanciant et en combinant de multiples ressources, qui seront des serveurs, des emplacements de stockage, des bases de données ou encore, des firewalls par exemple. Ainsi la virtualisation est nécessaire pour la mutualisation de ces ressources. Le rôle du cloud est l'automatisation, le suivi, et la facturation.
Les deux concepts permettent de maximiser les ressources informatiques et donc, de réduire les coûts et d'augmenter l'agilité de l'entreprise. **[8]**

VII. Principaux chalenges du Cloud :

Il existe encore beaucoup de problèmes qui ne sont pas complètement résolus dans le Cloud Computing. Dans cette section, nous mettrons en évidence certains des défis les plus importants pour l'avenir du Cloud Computing avec à leurs tête le problème de sécurité.

1. Les problèmes de sécurité:

La première et la principale préoccupation des clients du Cloud est le problème de sécurité. Comme leurs données sont stockées dans le nuage, les clients peuvent avoir de sérieuses raisons de s'inquiéter. Non seulement ils doivent faire confiance à un tiers pour la protection de leurs données, mais les clients ne peuvent pas localiser où leurs données sont stockées. Cela augmente leurs craintes face à des facteurs incertains tels que les règlements, les catastrophes, les conflits...

En plus des menaces extérieures, les données à l'intérieur du nuage sont également une cible potentielle de colocataires malveillants.

Les fournisseurs de cloud actuels tentent d'offrir une protection de sécurité adéquate aux petites et moyennes entreprises qui ne possèdent pas de solutions de sécurité sophistiquées. Dans le sondage sur la sécurité de l'information mondiale de 2012 de PwC **[16],** 54% des entreprises qui avaient adopté le nuage ont déclaré que le cloud améliorait réellement leur système de sécurité. Cependant, les entreprises ayant des données confidentielles élevées ne sont pas encore convaincues.

Il existe de nombreuses menaces dans le nuage, classées dans les catégories suivantes :

-Interfaces et API insécurisées: Les utilisateurs utilisent l'API pour communiquer avec le cloud et contrôler leurs machines et services virtuels. La transaction entre les utilisateurs et les serveurs API peut être exploitée pour voler les clés des utilisateurs ou les jetons d'accès, ou contourner la politique. Par conséquent, la protection des transactions API est essentielle pour le cloud.

-Fuite de données: passer de l'infrastructure privée de l'entreprise au nuage public nécessitera une attention particulière à la sécurité. Les exigences de sécurité doivent être adaptées à la nouvelle infrastructure réseau. Certaines données précédemment confinées sont maintenant exposées à Internet et partagées dans un réseau public. Même avec un nuage hybride où les données essentielles sont stockées chez les clients, le système informatique privé est exposé au nuage public, ce qui constitue une menace pour la sécurité. Ainsi, le Cloud Computing nécessite un contrôle d'accès étendu et un cryptage pour protéger les données privées.

-Confiance au fournisseur du Cloud: dans le cloud, des données de millions d'utilisateurs peuvent être hébergées par un seul fournisseur de cloud, et tous doivent lui faire confiance pour détenir leurs données. Les fournisseurs du Cloud doivent assurer la confidentialité des données de leurs clients. La politique de sécurité du fournisseur du Cloud doit inclure la mesure de la sécurité humaine, ex : Empêcher ses employés de tenter des actes malveillants. Certains fournisseurs de sites Web collectent des données auprès des clients pour la gestion et l'améliorent l'expérience des clients. Ces données, cependant, peuvent révéler les informations des clients et être exploitées pour d'autres activités avec ou sans sensibilisation des clients, ex : Publicité, spam ou simplement transfert vers des fournisseurs de services tiers.

-Clients malveillants: dans un environnement multi-tenant, de nombreux clients partagent les mêmes ressources. Le système de sécurité comprend la gestion de l'identité, le contrôle d'accès et l'isolement des clients d'autres clients et du système d'hypervision. Il existe le risque qu'un client malveillant tente de s'infiltrer dans l'espace d'autres clients ou dans le système d'hypervision. Certains systèmes de Cloud Computing utilisent la paravirtualisation pour améliorer les performances des machines virtuelles invitées qui peuvent également devenir une menace pour la sécurité. Par exemple, Rutkowska et son équipe ont réussi à compromettre Xen en utilisant l'attaque de BluePill **[17].**

-Utilisation abusive de Cloud Computing: en nuage public, les utilisateurs peuvent s'abonner en utilisant uniquement une carte de crédit valide; Certains nuages offrent une période d'essai. Les utilisateurs malveillants, les pirates informatiques et les spammeurs peuvent s'inscrire et mener leurs activités telles que l'hébergement de données malveillantes, l'envoi de spams, la récupération de données volées, le craque de mot de passe, le lancement d'attaques de déni de service, la commande de botnets, le lancement de points d'attaque dynamiques, etc.

-Le détournement de compte: comme tous les services en ligne, les comptes et les services des utilisateurs sont des cibles de détournement. L'hameçonnage, la fraude, l'exploitation des vulnérabilités des logiciels sont des menaces courantes.

-Compétence d'audit: les clients n'ayant aucun contrôle sur le système sous-jacent, ils ne peuvent pas connaître les détails du système, sa version de logiciel, sa mise à jour de code, sa conception de sécurité, etc. Le client a un accès limité aux journaux du réseau ainsi que la capacité de procéder à des enquêtes et collecte des données légales. Même pour le fournisseur de cloud, l'échelle du cloud rend difficile la vérification correcte. Certains fournisseurs de sites Web utilisent des ressources provenant de fournisseurs de cloud tiers, rendant ainsi l'audit encore plus difficile. **[16] [17]**

2. Standardisation et interopérabilité :

Dans un monde ouvert, les clients souhaitent choisir les fournisseurs de cloud qu'ils veulent et déplacer leurs systèmes / applications vers un autre quand ils en ont envi sans beaucoup d'efforts (portabilité).

Hélas, ce n'est pas encore le cas pour Cloud Computing. Les offres actuelles de Cloud Computing sont basées sur des ensembles d'architecture et d'API uniques. Même les outils dont le but est de créer un cloud privé compatible avec les offres publiques manquent encore de compatibilité pour travailler avec de nombreux nuages publics. La plupart d'entre eux sont basés sur l'API Amazon REST, qui offre la possibilité de passer à Amazon EC2, mais par ailleurs inutile lors du déplacement vers d'autres nuages publics tels que Joyent ou vCloud. Le système de construction qui s'étend de plusieurs nuages publics, par exemple entre Google AppEngine, Microsoft Azure, EngineYard, est presque impossible; et l'exportation de systèmes / applications construites dans un nuage public vers d'autres nuages publics (portabilité) n'est guère possible. Certains fournisseurs de petits sites rendent leur API compatible avec les grands clouds afin d'attirer les clients et de concurrencer les grands fournisseurs de cloud, mais dans l'ensemble, la migration d'application d'une plate-forme vers une autre plate-forme est impossible. Cela entraîne le problème du verrouillage des fournisseurs: le choix du fournisseur de cloud est permanent car la migration vers un autre fournisseur de cloud est presque impossible. Par exemple, force.com, bien que préconisé comme plate-forme PaaS ouverte, ne soit utile que pour la construction d'applications salesforce.com; ou les applications Cordy Process Factory qui sont construites à partir de l'assemblage d'autres composants personnalisés, ne fonctionnent que sur le nuage de Cordy.

Il existe plusieurs efforts de normalisation qui visent à fournir une interface commune et une interopérabilité entre les nuages. Toutefois, la technologie et le marché devront évoluer et se stabiliser un peu avant qu'une norme largement acceptée n'émerge. La responsabilité principale d'assurer le succès d'une telle norme incombe à la communauté des clients actuels et futurs de Cloud Computing. Certains clients et fournisseurs en nuage se regroupent en groupes tels que The Open Cloud Manifesto, The Open Group et Open Data Center Alliance pour fournir des cas d'utilisation et des défis à relever dans les efforts de normalisation.

Voici quelques normes de cloud:

-Interface de gestion des données en nuage (CDMI): CDMI ou Cloud Data Management Interface, définit une interface pour le stockage en nuage. Les fonctionnalités CDMI comprennent la création, la récupération, la mise à jour et la suppression de données, l'accès à la sécurité, la surveillance et la facturation depuis le cloud.

-Interface de gestion des infrastructures en nuage (CIMI): CIMI ou Cloud Infrastructure Management Interface, fournit une interface standard pour la gestion IaaS et permet de transférer les charges de travail des utilisateurs d'un nuage à l'autre.

-Open Cloud Computing Interface (OCCI): OCCI est une API extensible pour la gestion des nuages IaaS. Le noyau OCCI définit 3 catégories de ressources: calcul, réseau, stockage. L'Infrastructure OCCI ajoute l'interface réseau et le lien de stockage.

- Open Virtualization Format (OVF): OVF est un format ouvert extensible pour l'emballage et la distribution d'appareils virtuels (machines virtuelles et logiciels exécutés sur des machines virtuelles).

-Spécification de topologie et d'orchestration pour les applications en nuage (TOSCA): TOSCA ou Topology and Orchestration Specification for Cloud Applications, est conçu pour permettre l'interopérabilité des services de cloud d'infrastructure et d'application. TOSCA se concentre sur les relations entre les composants d'un service complexe et le comportement opérationnel de ces composants (par exemple, le déploiement, le patch, l'arrêt).

-Modèle d'utilisation Open Data Center Alliance: Open Data Center Alliance Usage Model est une série de modèles d'utilisation pour répondre aux défis, aux exigences et aux lignes directrices dans le cloud. Il contient actuellement 8 modèles d'utilisation dans quatre catégories: Secure Federation, Automation, Common Management & Policy et Transparency. **[18] [19] [20]**

3. Qualité de service:

Selon un sondage sur Micro Trend Inc **[22],** la qualité du service, en particulier la performance et la disponibilité des services en nuage, est la préoccupation numéro deux des clients en nuage, derrière le problème de sécurité. Les fournisseurs de cloud actuels n'offrent pas une qualité aussi élevée que l'informatique de l'entreprise. Cela rend difficile pour les entreprises commerciales de compter sur le cloud. Comme les instances de service continuent à augmenter, la probabilité qu'un échec d'instance se produise augmente de plus en plus. Les fournisseurs de Cloud essayent de s'attaquer au problème en fournissant des suivis de données à grains fins et en récupérant les pannes. Cependant, le redémarrage et la migration de VM nécessitent plusieurs secondes à plusieurs minutes, ce qui n'est pas acceptable dans l'informatique d'entreprise actuelle.

La concentration massive de ressources dans un fournisseur unique de cloud, généralement dans un datacenter, est en fait le seul point d'échec de l'ensemble du cloud. Étant donné que le fournisseur de nuage utilise uniquement une solution pour son service cloud, que se passe-t-il si cette solution présente un défaut ? Que se passe-t-il si quelqu'un réussit à pirater le système, ou un incident ou une catastrophe se produit? Les données du fournisseur de cloud, de leurs clients et des clients de ce dernier peuvent être perdues. Des milliers, des millions d'utilisateurs seront affectés par la grandeur que nous n'avons jamais vue auparavant. Le 2 janvier 2010, un périphérique de routage a provoqué une panne d'Amazon EC2 affectant de nombreux sites et plates-formes hébergés dans le centre de données de Virginie. Ce n'était pas la première panne dans les datacenters d'Amazon. En 2008, deux pannes ont eu lieu dans les services S3 en raison de la surcharge du service d'authentification et d'une erreur de bit unique. La même année, Google AppEngine a également eu 5 heures de panne partielle en raison d'une erreur de programmation.

Les fournisseurs de Cloud utilisent l'accord de niveau de service (SLA) pour documenter formellement la qualité du service (QoS), les attentes de performance, les responsabilités et les limites entre les fournisseurs de cloud et leurs clients. Un SLA typique décrit les niveaux QoS en utilisant différents attributs tels que la disponibilité du service, la durabilité des données, le temps de réponse et les pénalités associées aux violations de ces attributs. Le SLA doit être conçu correctement afin de réduire les conflits potentiels entre les fournisseurs de services et les clients en nuage. Une fois que le SLA est défini, les fournisseurs de services doivent pouvoir mesurer et surveiller les paramètres pertinents.**[16][22][21]**

4. Transfert de données lent:

Alors que la capacité de stockage est peu coûteuse et facile à acquérir, les ressources de liaison ne le sont pas. Même si nous pouvons installer plus de fibres pour augmenter la capacité de bande passante, la vitesse globalement élevée de l'Internet et le faible coût en sont encore loin. Imaginez combien de temps il faudrait pour transférer les données d'une entreprise vers le cloud ou transférer les données d'une entreprise d'un datacenter vers un autre datacenter.

Jim Gray, responsable du Centre de recherche de la région de Bay de Microsoft, a constaté que l'expédition de l'ensemble des disques de données ou des ordinateurs est beaucoup plus rapide que le transfert de données sur Internet. Prenons un exemple: supposons que nous voulons expédier 10 tonnes d'U.C. Berkeley à Amazon à Seattle, WA. La bande passante de l'écriture est mesurée à 5-18Mb / s. Faisons 20Mb / s, le transfert de données 10TB prendrait jusqu'à 4,000,000 secondes, ce qui équivaut à 45 jours. Si nous expédions le disque, il faudrait moins de un jour pour transférer 10 To, ce qui équivaut à 1.500Mb / s. Amazon propose actuellement un tel service appelé AWS Import / Export qui expédie le périphérique de stockage au datacenter Amazon. **[21][23]**

VIII. Les principaux systèmes IaaS :

Parmi les IaaS les plus repandus on retrouve :

1. OpenStack :

En 2010 la Nasa et Rackspace scellent un partenariat afin de fonder le projet OpenStack, qui vise à fournir un gestionnaire IaaS sous une licence de logiciel libre. L'objectif est de fournir une collection de services qui peuvent être utilisés indépendamment, mais qui ensemble, permettent la mise en place d'infrastructures de Cloud Computing. La première version d'OpenStack contenait deux services : le service Nova en charge de la gestion des machines virtuelles, dont le code venait de la plate-forme Nebula conçue par la Nasa, et le service Swift qui offre une solution de stockage évolutif dont le code était issu du projet Cloud File Platform utilisé par Rackspace.

OpenStack est une collection de services, où chaque service s'occupe d'un aspect précis d'une infrastructure IaaS. Ces services sont relativement faiblement couplés entre eux, permettant des déploiements modulaires où seuls les services basiques sont indispensables.

Par sa volonté de fournir un gestionnaire IaaS qui soit modulaire et modifiable, et grâce à sa licence logicielle libre (Apache 2.0), le projet OpenStack a connu un succès rapide, comptant en

janvier 2016 prés de 1800 contributeurs (tous projets confondus), et est utilisé par une variété d'acteurs privés tels que Rackspace, Ebay, Yahoo, HP, Orange, ainsi que des institutions publiques ou académiques telles que le MIT, le Argonne National Laboratory, le CERN et l'Université de Strasbourg. **[10]**

2. Cloudstack :

CloudStack est une plate-forme open source IaaS Cloud développée à l'origine par Cloud.com. En avril 2012, Citrix a fait don de CloudStack à Apache Software Foundation, tout en changeant la licence pour Apache 2.0. CloudStack implémente les API Amazon EC2 et S3, ainsi que l'API vCloud, en plus de son propre API. CloudStack, écrit en Java, est conçu pour gérer et déployer de grands réseaux de machines virtuelles. CloudStack supporte actuellement VMware, Oracle VM, KVM, XenServer et Xen Cloud Platform. CloudStack a une structure hiérarchique, qui permet la gestion de plusieurs hôtes physiques à partir d'une seule interface. **[11][12]**

3. Eucalyptus :

En 2009 des chercheurs de l'Université de Californie publient "The Eucalyptus Opensource Cloud-computing System" qui s'intéresse aux problèmes inhérents au modèle des grilles de calcul : les auteurs analysent que pour les très gros besoins en ressources de calcul, il est nécessaire d'agréger les ressources de plusieurs fournisseurs, ce qui se traduit bien souvent par une hétérogénéité matérielle. Cette hétérogénéité doit être prise en compte par les concepteurs de service, ce qui nécessite des compétences techniques avancées.

Les auteurs proposent alors de mettre en place un accès uniforme à cette collection de ressources, en les fournissant aux utilisateurs sous la forme de machines virtuelles. L'infrastructure sur laquelle sont déployées ces machines virtuelles est gérée par le système Eucalyptus qui est défini par ses auteurs comme un framework logiciel permettant de transformer des ressources de calcul et de stockage d'une organisation en ressources de Cloud Computing. **[13]**

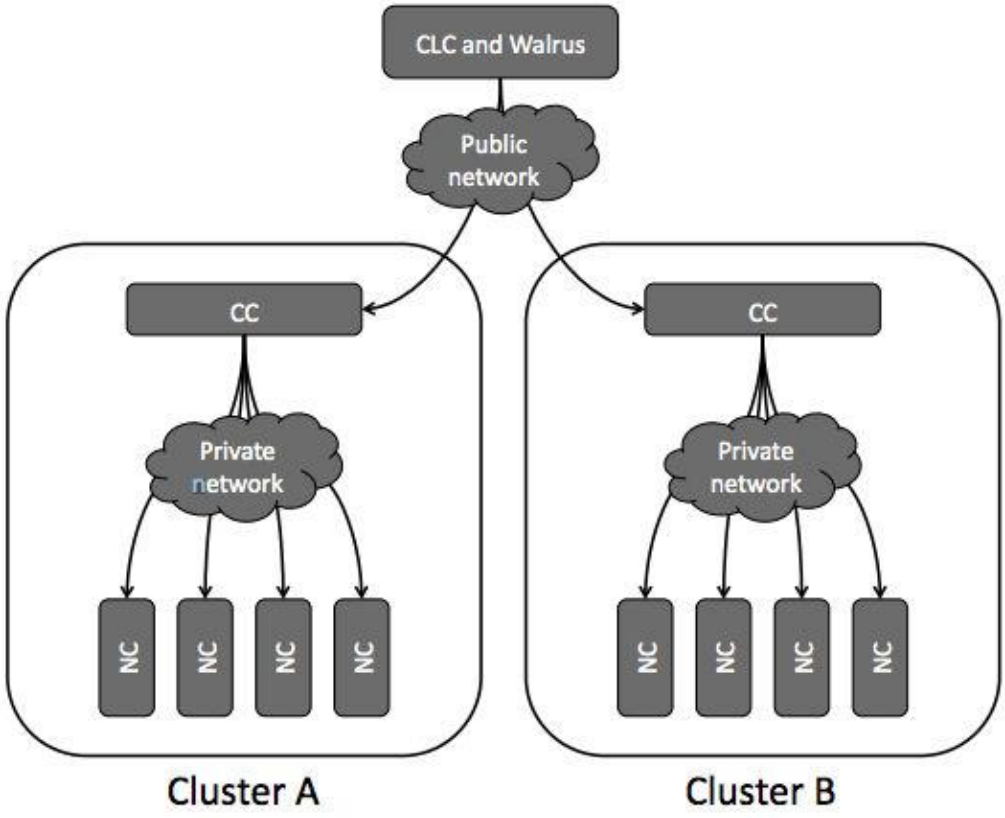

Figure 2: Architecture du gestionnaire Eucalyptus. [13]

Le système Eucalyptus est un gestionnaire IaaS conçu autour d'une architecture modulaire, où chaque composant est responsable d'une fonction précise de l'infrastructure IaaS, tout en restant indépendant des autres composants. Chacun des composants d'Eucalyptus est rendu accessible aux autres composants sous la forme d'un service web. Une infrastructure IaaS, utilisant Eucalyptus, est organisée de façon hiérarchique : une infrastructure de grande taille sera divisée en plusieurs groupes de noeuds (clusters) de tailles inférieures et où chaque serveur de l'infrastructure est spécialisé comme indiqué par la figure 2:

• **Cloud Controller :** sert de point d'entrée pour communiquer avec l'infrastructure IaaS. Il expose une API web qui peut être utilisée pour manipuler l'infrastructure sous-jacente.

• **Cluster Controller :** est déployé dans chacun des clusters de l'infrastructure. Ce type de noeud surveille l'exécution des machines virtuelles au sein du cluster dont il est responsable. Il reçoit des ordres de création de machines virtuelles du Cloud Controller et s'occupe de l'ordonnancement en choisissant le Node Controller qui les hébergera.

• **Node Controller :** est responsable du contrôle et de la surveillance des machines virtuelles qu'il héberge. Il reçoit des ordres de création de machines virtuelles du Cluster Controller.

• **Walrus (Storage Controller) :** est en charge de la fourniture d'un service de stockage utilisant les mêmes interfaces logicielles que le service Amazon S3. Walrus est principalement utilisé pour le stockage des images des machines virtuelles ainsi que les métadonnées des utilisateurs.

Concernant les communications entre machines virtuelles, Eucalyptus propose de connecter ces dernières à travers un système de réseau virtuel, défini de manière logicielle en se basant sur les techniques de réseau d'overlay (Overlay Network). Cela permet à des machines virtuelles qui n'appartiennent pas à la même organisation d'être isolées.

Cette technique de virtualisation du réseau entrainant un léger surcoût (un saut réseau supplémentaire via l'ordinateur hébergeant la machine virtuelle.), le choix est donc laissé à l'utilisateur entre privilégier l'isolation de ses ressources de calculs ou de reposer sur un réseau plus conventionnel.

Enfin, Eucalyptus est programmable via une interface logicielle (API) qui permet à ses utilisateurs de manipuler une infrastructure déployée avec Eucalyptus. Celle-ci est calquée sur les interfaces proposées par les services phares d'Amazon EC2 et S3. Cela permet au projet Eucalyptus d'avoir l'ambition d'être utilisé sans modification par des outils reposants sur les deux services évoqués précédemment.

4. OpenNebula :

Parallèlement à l'apparition du projet Eucalyptus, le projet OpenNebula émerge en 2009 et est le fruit de la collaboration de chercheurs issus de l'université de Chicago, de l'université de Madrid ainsi que du laboratoire d'Argonne proposent le projet OpenNebula. Ce projet vise à fournir un gestionnaire d'IaaS misant sur l'ouverture à la fédération avec des infrastructures IaaS tierces.

Faisant le constat qu'en 2009 l'écosystème des infrastructures de Cloud Computing se partage entre les Clouds publics et les Clouds privés (et Clouds hybrides), et que ces derniers occupent une place de plus en plus importante, l'équipe du projet OpenNebula estime qu'il faut rendre possible et même faciliter la mise en place d'infrastructures IaaS qui soient exploitées par plusieurs fournisseurs. Dans cette optique, le projet OpenNebula propose un gestionnaire IaaS qui se place à un niveau d'abstraction plus élevé que les projets d'alors tels Nimbus et Eucalyptus, afin de faciliter la fédération d'infrastructures appartenant à plusieurs fournisseurs.

L'absence de standards ouverts pose le problème de la fédération d'infrastructures qui ne possèdent pas les mêmes APIs, voire qui ne partagent pas toutes les mêmes fonctionnalités.

OpenNebula a été conçu pour être nativement ouvert à l'utilisation de ressources issues de fournisseurs externes, grâce à un système de pilotes logiciels externes (Cloud Drivers). Cette approche permet à OpenNebula de pouvoir piloter les ressources de calculs, qu'elles soient hébergées localement ou dans des infrastructures distantes, du moment que ces dernières sont rendues disponibles via un Cloud Driver. Initialement, les infrastructures externes supportées étaient Amazon EC2, Eucalyptus et ElasticHosts. **[14]**

La figure 3 résume les différents services composant le gestionnaire OpenNebula :

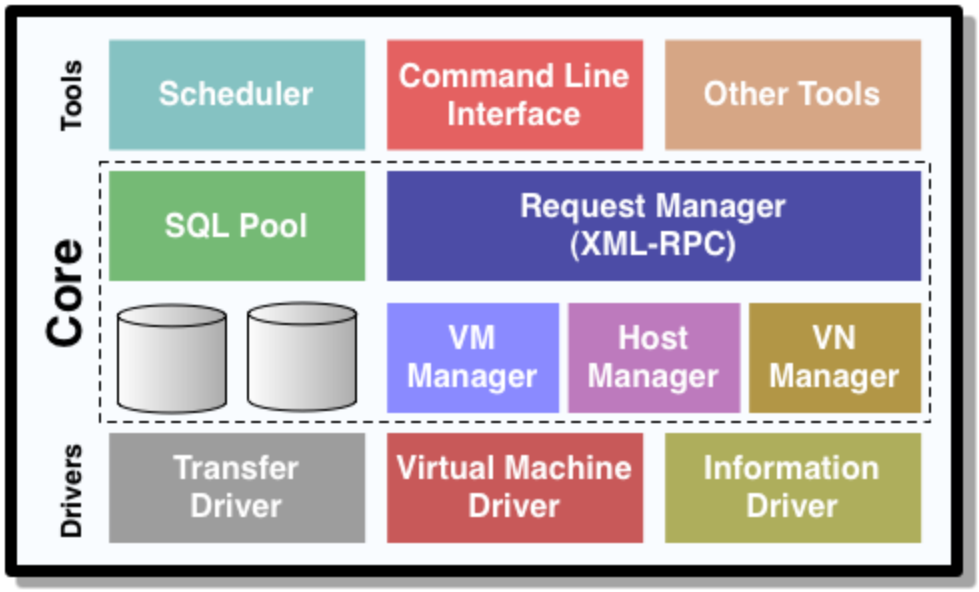

Figure 3: Architecture du projet OpenNebula. [15]

IX. Comparaison entre différentes solutions IaaS :

Actuellement, plusieurs solutions de cloud computing IaaS existent. Les administrateurs, les développeurs et les utilisateurs doivent choisir le meilleur environnement pour eux. La disponibilité de plates-formes Cloud libres open source est essentielle pour stimuler davantage la prolifération des environnements cloud et hybride privés. Dans cette partie, nous présentons les résultats d'une étude comparative **[12]** entre les IaaS présentés dans le point précédent pour sélectionner le meilleur pour le déploiement et le développement de la recherche que nous voulons mener.

	Solutions Iaas de Cloud Computing			
	Eucaliptus	OpenNebula	CloudStack	OpenStack
Stockage	+++++	+++	+++++	+++++
Réseau	++++	++++	+++++	+++++
Sécurité	++++	+++	++++	+++++
Hyperviseur	++++	+++	+++++	++++
Evolutif	+++	++++	+++++	+++++
Documentation	+++	+++	+++++	+++++
Installation	++	+++	+++++	+++++
Open source	+++	+++++	+++++	+++++

Tableau 1: Comparaison des diffentes solutions IaaS. [12]

Selon cette étude comparative, il semble que la meilleure solution soit OpenStack, elle peut devenir la solution de référence du cloud computing open source en raison de ses caractéristiques. Ceci nous motive à adopter OpenStack pour notre environnement et à le voir plus en détails dans le chapitre III.

Conclusion:

Dans ce chapitre nous avons introduit et défini la notion de cloud computing, tout en donnant un état de l'art autours de ce concept. L'étude comparative effectuée en fin de chapitre nous a permit de décider quel solution IaaS utiliser pour mener à bien notre travail.

Chapitre II :

La Sécurité informatique, Honeypots et Honeynets.

Introduction:

La sécurité informatique est de nos jours devenue un problème majeur dans la gestion des réseaux d'entreprise ainsi que pour les particuliers de plu en plus nombreux à se connecter à Internet. La transmission et la sauvegarde d'informations sensibles et le désir d'assurer la sécurité de celles-ci est devenue un point primordial dans la mise en place de réseaux informatiques.

Dans ce chapitre nous allons étudier aussi bien les motivations et les techniques utilisées par les pirates informatiques que les outils de sécurité qui servent à les contrer. Nous allons aussi voir plus en détails la tactique de sécurité choisie pour la suite de notre travail qui est les Honeypots.

I. Sécurité d'un réseau

La sécurité d'un réseau est un ensemble de moyens techniques, organisationnels, juridiques et humains mis en place pour conserver, rétablir, et garantir la sécurité des différents composants du réseau. En général, la sécurité d'un réseau englobe celle du système informatique sur lequel il s'appuie. **[25]**

II. Évaluation de la sécurité d'un réseau:

La sécurité d'un réseau peut s'évaluer sur la base d'un certain nombre de critères de sécurité. On distingue généralement cinq principaux critères de sécurité : **[25][36]**

• **Disponibilité :** Permettant de maintenir le bon fonctionnement du système. Elle consiste à garantir l'accès à un service ou à une ressource lorsqu'un utilisateur autorisé en a besoin.

• **Intégrité :** Elle consiste à s'assurer que les données n'ont pas été altérées durant la communication (de manière intentionnelle ou non) c'est-à-dire garantir que les données sont bien celles que l'on croit être.

• **Confidentialité :** Elle consiste à rendre l'information inintelligible à d'autres personnes que les seuls acteurs concernés.

• **Authentification :** Elle consiste à assurer l'identité d'un utilisateur, c'est-à dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe) l'accès à des ressources uniquement aux personnes autorisées.

• **Non répudiation :** Elle garantie qu'aucun des correspondants ne pourra nier la transaction.

III. Raisons qui poussent à la sécurisation des réseaux :

Nombreuses sont les raison qui peuvent nous pousser à sécuriser un réseau parmi elles :

1. Les enjeux :

Les différents enjeux sont : **[25]**

a) **Enjeux économiques :** Les organismes ou entreprises à but lucratif on presque toujours la même finalité : c'est de réaliser des bénéfices sur l'ensemble de leurs activités. Cette réalisation est rendue possible grâce à leurs systèmes d'informations considérés comme moteurs de développement des entreprises. D'où la nécessité de garantir la sécurité de ces derniers. La concurrence fait que des entreprises investissent de plus en plus dans la sécurisation de leurs systèmes d'information.

b) **Enjeux politiques :** La plupart des entreprises ou organisations se réfèrent aux documents officiels de sécurité élaborés et recommandés par l'État. Ces documents contiennent généralement des directives qui doivent être appliquées par toute structure engagée dans un processus de sécurisation du réseau. Dans le cadre du chiffrement des données par exemple, chaque État définit des cadres et mesures d'utilisation des algorithmes de chiffrement et les recommande aux entreprises exerçant sur son territoire. Le non respect de ces mesures et recommandations peut avoir des conséquences graves sur l'entreprise. A ce niveau, l'enjeu est plus politique parce que chaque État souhaite être capable de décrypter toutes les informations circulant dans son espace.

c) **Enjeux juridiques :** Dans un réseau, on retrouve de l'information multiforme (numérique, papier, etc.). Le traitement de celle ci doit se faire dans un cadre bien définit et dans le strict respect des lois en vigueur. En matière de juridiction, le non respect des lois et exigences relatives à la manipulation des informations dans un système d'information peut avoir des conséquences graves sur l'entreprise.

2. Les vulnérabilités :

Tous les systèmes informatiques sont vulnérables. Cependant le niveau de vulnérabilité diffère d'un système à un autre. Une vulnérabilité est une faille ou une faiblesse pouvant être exploitée par une personne mal intentionnée pour nuire. Les vulnérabilités des systèmes peuvent être classées en trois catégories : **[25]**

a) Vulnérabilités humaines : L'ingénierie sociale est la technique de craquer login, mot de passe ou d'autres informations importantes par la tromperie. Un pirate pourrait appeler un employé, se présentant comme un collègue, et demander le mot de passe du réseau ou d'autres informations. En outre, les utilisateurs peuvent propager des programmes malveillants sur les systèmes protégés, provoquant une brèche de sécurité. Une formation de base à la sécurité pour les utilisateurs et pour le personnel peut aider à réduire cette vulnérabilité.

b) Vulnérabilités technologiques : Avec la progression exponentielle des outils informatiques, des vulnérabilités technologiques sont découvertes tous les jours. Ces vulnérabilités sont à la base dues à une négligence humaine lors de la conception et la réalisation. Pour être informé régulièrement des vulnérabilités technologiques découvertes, il suffit de s'inscrire sur l'une des listes de diffusion mises en place par les CERT (Computer Emergency Readiness ou Response Team).

c) Vulnérabilités organisationnelles : Les vulnérabilités d'ordre organisationnel sont dues à l'absence de documents formels, de procédures de travail et de validation suffisamment détaillées pour faire face aux problèmes de sécurité du système. Quand bien même ces documents et procédures existent, leur vérification et mises à jour ne sont pas toujours bien assurées.

3. Les Menaces :

Les menaces peuvent être classées en deux catégories : **[26]**

a) **Les menaces passives :** consistent essentiellement à copier ou à écouter l'information sur le réseau, elles nuisent à la confidentialité des données. Dans ce cas, celui qui prélève une copie n'altère pas l'information elle même.

b) **Les menaces actives :** sont de nature à modifier l'état du réseau.

4. Les Risques :

Le risque en termes de sécurité est généralement caractérisé par l'équation suivante :

$$Risque = \frac{Vulnérabilité\ X\ menace}{Contre-mesure}$$ **[36]**

La vulnérabilité et la menace étant expliquées précédemment, la contre-mesure, est l'ensemble des actions mises en œuvre en prévention de la menace. Les contre-mesures à mettre en œuvre ne sont pas uniquement des solutions techniques mais également des mesures de formation et de sensibilisation à l'intention des utilisateurs, ainsi qu'un ensemble de règles clairement définies. **[26] [27]**

IV. Pirates et attaques informatique :

Ce point tourne autours des attaques qui menacent les systèmes informatiques ainsi que leurs buts.

1. Haker

Selon wikipedia « Hacker » est un mot d'origine anglo-américaine. Dans son sens général, un hacker est quelqu'un qui aime comprendre le fonctionnement d'un mécanisme, afin de pouvoir le modifier pour le détourner de son fonctionnement originel. Appliqué à l'informatique, un hacker sait où et comment manier un programme ou matériel électronique pour effectuer des tâches autres que celles prévues par ses concepteurs.

Il existe deux types de Hackers : celui qui essai de s'infiltrer dans un système pour prouver qu'il est capable d'y prendre des informations et de les modifier. Aussi appelé white hat, il ne détruit rien. Il avertit toujours les propriétaires des systèmes en question des failles qu'il a trouvées. Contrairement au second type, dit aussi cracker (Pirate) ou encore black hat. Sans les white hat, l'informatique et internet serait bourré de failles et de bugs.

2. Les logiciels malveillants :

Un logiciel malveillant est un programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur. Plusieurs types de logiciels malveillants ont été proposés nous citons les plus répandus :

a. Virus :

Un virus est un morceau de programme informatique malicieux, conçu et écrit pour qu'il se reproduise. Cette capacité à se répliquer, peut toucher un ordinateur, sans la permission du propriétaire et sans que ce dernier ne le sache. En termes plus techniques, le virus classique s'attachera à l'un des programmes exécutables et se copiera systématiquement sur tout autre exécutable lancé. Les virus peuvent être particulièrement dangereux et endommager plus ou moins gravement les machines infectées. Le virus peut se répandre à travers tout moyen d'échange de données numériques comme l'Internet, et notamment par l'intermédiaire des messages électroniques ou de leurs pièces attachées. **[28]**

b. Vers :

Un ver (ou worm) est un type de virus particulier qui se propage par le réseau. Les vers contrairement aux virus, une fois implantés et activés dans un ordinateur, sont des programmes capables de se propager d'un ordinateur à un autre via le réseau, sans intervention de l'utilisateur et sans exploiter le partage de fichiers. **[28]**

c. Cheval de Troie :

Un cheval de Troie (Trojan horse) est un programme qui exécute des instructions sans l'autorisation de l'utilisateur. Ces instructions sont généralement nuisibles à l'utilisateur, et qui une fois installé sur un ordinateur y effectue des actions cachées. Le cheval de Troie contrairement au ver ne se réplique pas. **[29]**

d. Logiciel Espion :

Un logiciel espion (ou spyware) est un programme, conçu dans le but de collecter des données personnelles sur ses utilisateurs et de les envoyer à son concepteur, ou à un tiers via Internet ou tout autre réseau informatique, sans avoir obtenu au préalable une autorisation explicite et éclairée des dits utilisateurs. Une variété particulièrement toxique de logiciel espion est le keylogger (espion dactylographique), qui enregistre fidèlement tout ce que l'utilisateur tape sur son clavier et le transmet à son honorable correspondant. Il capte ainsi identifiants, mots de passe et codes secrets. **[29] [25]**

e. Spam :

Le spam est une vraie problématique. Il encombre les résultats de recherche ce qui gène l'utilisateur. Un spam peut être définit comme étant un email anonyme, non sollicité, indésirable et envoyé en grand nombre de façon automatique sans l'accord de son destinataire. **[27]**

f. Cookies :

Un cookie est un petit fichier texte, enregistré sur le disque dur de l'ordinateur d'un internaute à la demande du serveur gérant le site Web visité. Il contient des informations sur la navigation effectuée sur les pages de ce site. L'idée originelle est de faciliter l'utilisation ultérieure du site par la même personne. Un cookie n'étant pas exécutable, il ne peut contenir de virus. **[27]**

g. Bombe logique :

Une Bombe logique est une partie d'un programme malveillant (virus, cheval de Troie, etc.) qui reste dormante dans le système hôte jusqu'à ce qu'un instant ou un événement survienne, ou encore que certaines conditions soient réunies, pour déclencher des effets dévastateurs en son sein. Les bombes logiques sont généralement utilisées dans le but de créer un déni de service en saturant les connexions réseau d'un site, d'un service en ligne ou d'une entreprise. **[30]**

h. Porte dérobée :

C'est un moyen de contourner les mécanismes de contrôle d'accès. Elle s'agit d'une faille du système de sécurité due à une faute de conception accidentelle ou intentionnelle. **[27]**

3. Motivation des attaques :

Les motivations des attaques peuvent être différentes en voici quelques-unes :

• Obtenir un accès au système ;
• Voler des informations, tels que des secrets industriels ou des propriétés intellectuelles ;
• S'informer sur l'organisation (entreprise de l'utilisateur, etc.) ;
• Troubler le bon fonctionnement d'un service ;
• Utiliser le système de l'utilisateur comme « rebond » pour une attaque ;
• Utiliser les ressources du système de l'utilisateur.

V. Mécanismes de sécurité :

À cause des menaces provenant des logiciels malveillants, Il faut mettre en place des mécanismes pour s'assurer de la confidentialité, l'intégrité et la disponibilité des services. Parmi ces mécanismes, on peut citer :

1. Cryptage :

Le chiffrement est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de (dé)chiffrement. Ce principe est généralement lié au principe d'accès conditionnel. Bien que le chiffrement puisse rendre secret le sens d'un document, d'autres techniques cryptographiques sont nécessaires pour communiquer de façon sûre. **[33]**

2. Pare-feu :

C'est un ensemble de différents composants matériels (physique) et logiciels (logique) qui contrôlent le trafic intérieur/extérieur selon une politique de sécurité. Un système pare-feu fonctionne la plupart du temps grâce à des règles de filtrage indiquant les adresses IP autorisées a communiquer avec les machines aux réseaux, il s'agit ainsi d'une passerelle filtrante. Il permet d'une part de bloquer des attaques ou connexions suspectes d'accéder au réseau interne. D'un autre côté, un firewall sert dans de nombreux cas également à éviter la fuite non contrôlée d'informations vers l'extérieur. Il propose un véritable contrôle sur le trafic réseau de l'entreprise, Il permet donc d'analyser, de sécuriser et de gérer le trafic réseau. **[35]**

3. Antivirus :

Les antivirus sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants. Ceux-ci peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de programmes modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur de l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur. Un antivirus vérifie les fichiers et courriers électroniques, les secteurs de boot (pour détecter les virus de boot), mais aussi la mémoire vive de l'ordinateur, les médias amovibles (clefs USB, CD, DVD, etc.), les données qui transitent sur les réseaux (dont internet), etc. **[31]**

4. VPN :

Dans les réseaux informatiques, le réseau privé virtuel (Virtual Private Network en anglais, abrégé en VPN) est une technique permettant aux postes distants de communiquer de manière sûre, tout en empruntant des infrastructures publiques (internet). Un VPN repose sur un protocole, appelé protocole de tunnelisation, c'est-à-dire un protocole permettant aux données passant d'une extrémité à l'autre du VPN d'être sécurisées par des algorithmes de cryptographie. **[34]**

5. IDS :

Un système de détection d'intrusion (ou IDS : Intrusion Detection System) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions. Il faut distinguer deux aspects dans le fonctionnement d'un IDS : le mode de détection utilisé et la réponse apportée par l'IDS lors de la détection d'une intrusion. **[35]**

6. IPS :

Un système de prévention d'intrusion (ou IPS, Intrusion Prevention System) est un outil similaire aux IDS, sauf que ce système peut prendre des mesures afin de diminuer les risques d'impacts d'une attaque. C'est un IDS actif, il détecte un balayage automatisé, l'IPS peut bloquer les ports automatiquement. **[35]**

VI. Définition d'un Honeypot :

Un Honeypot (pot de miel) est une machine, qu'on fait exprès de ne pas trop sécuriser et où l'on ne stocke pas de données personnelles ou importantes, voir même où l'on va mettre de fausses informations.

Cette machine servira de leurre pour les pirates, qui se feront un plaisir à l'attaquer et à récolter les données s'y trouvant. L'intérêt de doter un Honeypot d'une faible sécurité est de faciliter son accès aux attaquants sans pour autant éveiller leurs soupçons. De cette manière, chaque fait commit par un hacker sera récolté et pourra donc par la suite être analysé pour savoir quel types d'attaques ont été utilisées, et ainsi pouvoir mettre en place de meilleures stratégies de défense.

L'installation d'un pot de miel peut avoir plusieurs objectifs : **[39]**

- Analyser le comportement d'un pirate ou d'un logiciel malveillant ;
- Identifier les postes compromis par des malwares réalisant des scans du réseau interne pour se propager ;
- Identifier les malwares qui se propagent sur un réseau;
- Observer les types d'attaques menées par les pirates ;
- L'un des buts principaux est d'éloigner l'intrus du système de production.

VII. Historique des Honeypots:

L'idée n'est pas nouvelle, en 1986 lorsqu'un administrateur système cherchait à comprendre pourquoi il avait 75 % de perte de données au niveau de la comptabilité, Il découvrit un pirate informatique sérieusement ancré dans le réseau qui espionnait et revendait ses informations. Il décida alors de fabriquer de faux fichiers et réussit ainsi à leurrer l'attaquant pendant un certain temps, temps nécessaire pour le localiser et le faire arrêter.

Le 7 Janvier 1991 Bill Cheswick, un programmeur système spécialisé dans la sécurité et travaillant chez AT&T Bell, a commencé à jouer avec un pirate en répondant manuellement à ces requêtes qui tentaient d'exploiter une faille dans le démon sendmail. Le programmeur a alors réussi à lui faire croire qu'il avait obtenu une copie du fichier passwd qui contenait en réalité que des comptes falsifiés et contrôlés par le programmeur de AT&T Bell. Il a alors laissé le pirate s'amuser sur la machine et a étudié son comportement pendant plusieurs mois. Il a ensuite publié un papier sur ses découvertes. **[42]**

Ce n'est seulement qu'en 2000 que le terme honeypot a vraiment été intégré dans le monde des entreprises. En effet, c'est à cette époque que l'on a vu le déploiement des premiers honeypots.

Depuis 2012 les honeypots commencent à être adoptés comme solution de sécurité dans les environnements de Cloud Computing. **[38]**

VIII. Caractéristiques d'un honeypot :

A partir de la définition, on peut déduire les caractéristiques suivantes **[39]** :

- Un honeypot est un système qui n'a aucune valeur métier/business en terme de production ;
- L'ensemble des communications relatives aux honeypots peut être considéré comme étant malveillant : un honeypot cherchant à se connecter à une autre ressource est probablement compromis et il n'y a aucune raison qu'un utilisateur légitime interagisse avec ce système ou ce service;
- Un honeypot est à la fois un trompe-l'oeil et un piège pour les attaquants : ces derniers perdent leur temps en s'y attaquant, et leurs actions sont minutieusement épiées ;
- Un honeypot ne peut pas empêcher une attaque comme le pourrait un IPS ou un firewall. Cependant, ce type d'outil permet de la détecter, ainsi que de détecter ses principales caractéristiques : cible, origine, technique d'exploitation utilisées, ... Un pot de miel devrait donc, dans l'idéal, être utilisé de manière conjointe avec un pare-feu ou un IDS, de manière à protéger ou plutôt à alerter de la possible malveillance interne.

IX. Les types de Honeypots :

Les honeypots peuvent être classés selon plusieurs critères de classification : **[40] [41] [43]**

1. Selon le Niveau d'interaction :

Décrit si le honeypot est limité dans la façon dont il expose ses fonctionnalités.

a. Honeypots à faible interaction :

Ils sont les plus simples de la famille des honeypots. Leur but est de récolter un maximum d'informations tout en offrant un minimum de privilèges aux pirates. Ils permettent de limiter les risques au maximum. Contrairement à un honeypot à forte interaction, il ne fait que simuler ces services et ne les possède pas réellement comme montré dans la figure 4. Ils ne peuvent donc pas être exploités par les malwares pour se déployer. Ces types de honeypots sont faciles à mettre en place et à maintenir. L'administrateur a le contrôle total sur le processus d'intrusion. Mais ils sont faciles à détecter par les intrus, dépendamment de l'efficacité de leur simulation. Et les informations récoltées ne sont pas forcément pertinentes.

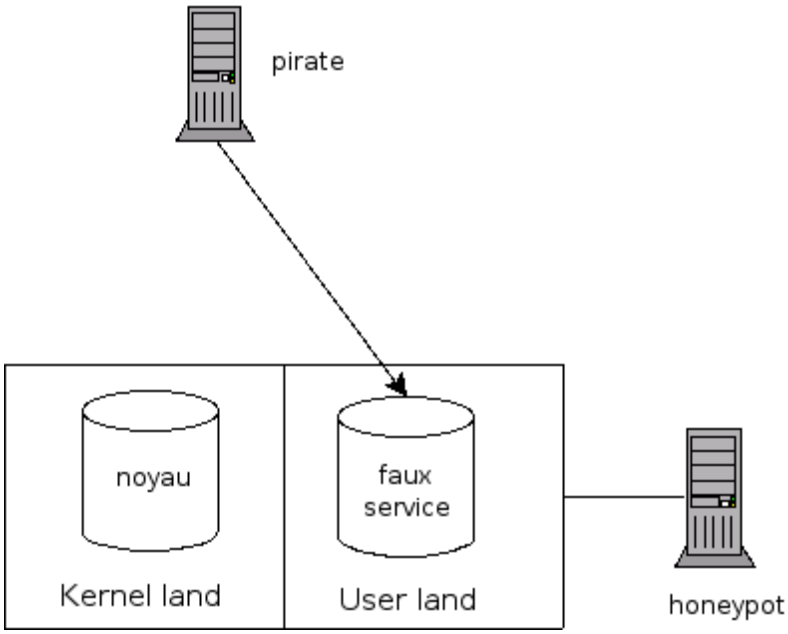

Figure 4: Honeypot à faible interaction. [38]

b. Honeypots à forte interaction :

Ce type de honeypots peut être considéré comme le côté extrême du sujet puisqu'il repose sur le principe de l'accès à de véritables services sur une machine du réseau plus ou moins sécurisée, voir figure 5. Cela permet de récupérer beaucoup d'informations pertinentes et la découverte de nouveaux types d'attaques. Mais ce type de honeypots est durs à configurer et à maintenir.

Les risques sont beaucoup plus importants dans les honeypots à forte interaction que pour les honeypots à faible interaction. Il apparaît donc nécessaire de sécuriser au maximum l'architecture du réseau pour que l'attaquant ne puisse pas rebondir et s'en prendre à d'autres machines.

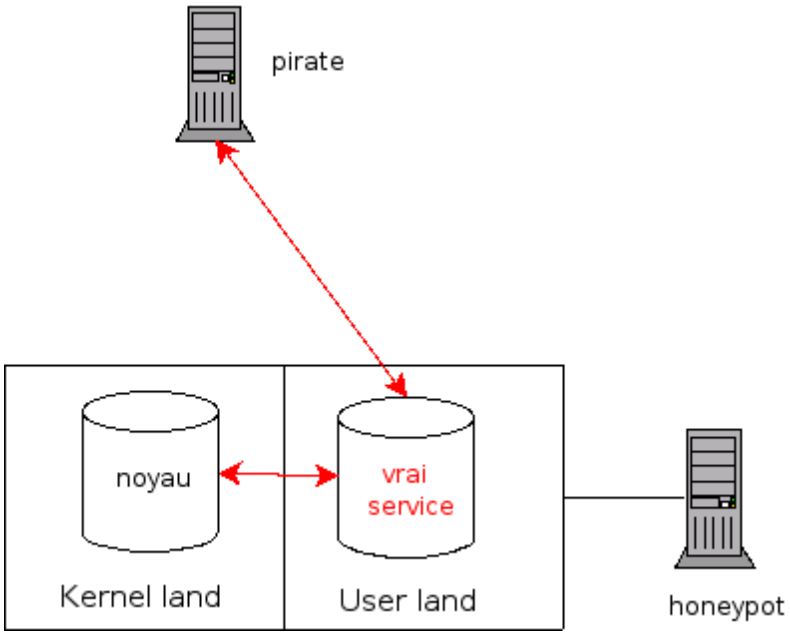

Figure 5: Honeypot à forte interaction. [38]

c. Honeypots hybrides :

Les honeypots hybrides combinent à la fois des outils à faible interaction et des outils à forte interaction afin d'en tirer le plus d'avantages possible.

Dans SGNET (Un cadre déployable mondial pour soutenir l'analyse des modèles de menace de Malware) **[40],** un honeypot côté serveur à interaction élevée est utilisé pour apprendre à gérer le trafic inconnu. Ex. Comment émuler de nouveaux protocoles. Après ce processus d'apprentissage, un trafic similaire est redirigé vers des honeypots à faible interaction. Cette combinaison augmente à la fois le niveau de détection des menaces et la performance.

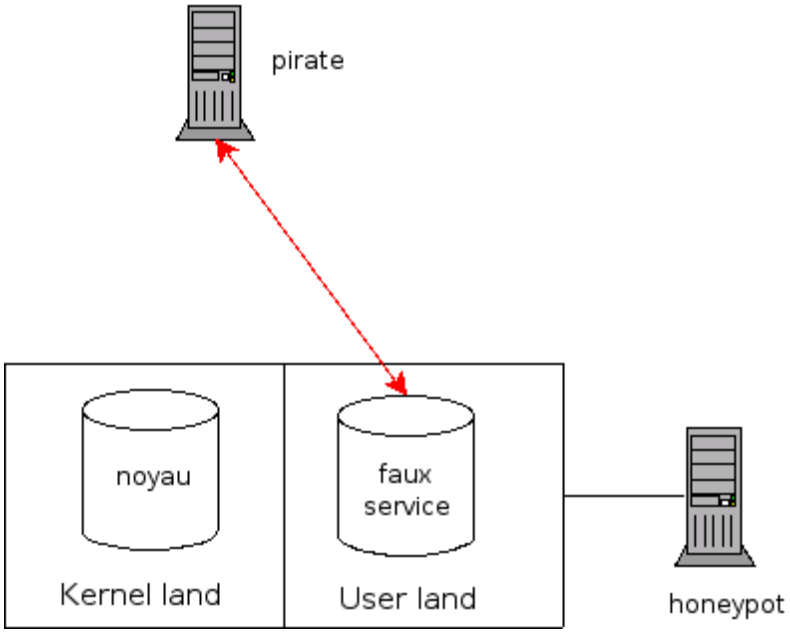

Figure 6: Honeypot à moyenne interaction. [38]

2. Selon le type de ressources attaquées :

Décrit si les ressources d'un honeypot sont exploitées en mode serveur ou client.

a. Honeypots coté server :

Ce sont des Honeypots conçus pour détecter et étudier les attaques sur les services réseau. Les Honeypots de ce type agissent comme un serveur. Ils exposent un port ouvert, plusieurs ports ou des applications entières et écoutent passivement les connexions entrantes, établies par des clients distants (probablement malveillants). Souvent, ce type de honeypots détecte les menaces qui utilisent le scan comme moyen d'identifier les victimes potentielles à compromettre, par exemple : l'analyse des vers ou des bots. Mais ils peuvent également être utilisés pour détecter les tentatives manuelles d'accès aux machines. Les honeypots côté serveur sont considérés comme des honeypots «traditionnels», et le terme «honeypots» est par défaut associé à eux.

Les honeypots coté server peuvent être divisés en sept sous-classes, selon quel service ou technique d'attaque / détection est implémentée sur un honeypot donné :

- **Honeypots d'applications Web:** outils de détection d'attaques sur des applications Web,
- **Honeypots SSH:** outils orientés sur les attaques Secure Shell (SSH)
- **Similaires SCADA:** outils d'émulation de systèmes de contrôle industriel,
- **VoIP honeypots:** outils détectant les menaces dans la téléphonie sur Internet (voix sur IP),
- **Bluetooth honeypots**: outils visant à détecter les attaques propagées via la technologie Bluetooth,
- **USB honeypots:** outils visant à détecter les attaques utilisant des périphériques USB,
- **Sinkholes:** outils à l'aide d'une technique «sinkhole» pour détecter et surveiller les infections dans un réseau,
- **Honeypots à usage général:** outils visant à détecter plus d'une technique d'attaque ou plus d'un service.

b. Honheypots coté client :

Dits aussi « Honeyclients », ce sont des Honeypots conçus pour détecter les attaques sur les applications clientes. Les Honeypots côté client sont très différents dans leur fonctionnement des Honeypots coté serveurs. Un Honeyclient établit activement des connexions aux services afin de détecter les comportements malveillants soit du serveur, soit du contenu qu'il dessert. Les Honeyclients les plus populaires sont ceux qui détectent les attaques sur les navigateurs Web et leurs plugins, propagés via des pages Web. Certains ont également la capacité de supporter diverses formes de pièces jointes.

c. Honeytokens :

Un honeytoken est une ressource stockée ou traitée par un système informatique (par exemple: un fichier texte, un message électronique ou un enregistrement de base de données) qui ne peut être récupéré dans des conditions normales dans un environnement de production. En d'autres termes, tout accès aux données Honeytoken devrait être considéré comme une action malveillante.

Il existe également d'autres classifications d'après Christian Seifert, Ian Welch and Peter Komisarczuk **[41]** qui sont :

3. Selon l'environnement de mise en place:

Les honeypots sont essentiellement utilisés dans des environnements comme outils de production ou de recherche.

a. Production :

Dans ce cas, le honeypot est utilisé comme moyen de protection et de défonce. Dans un environnement de production, un honeypot est utilisé pour dérouter les attaques vers des machines leurres qui simulent les services d'un réseau de production. Ce type d'honeypot ne requiert pas beaucoup de vigilance de la part de l'administrateur. Il doit uniquement surveiller les intrusions sur les machines leurres afin de s'assurer que les failles exploitées ne sont pas présentes sur les machines de production réelles.

b. Recherche :

C'est le domaine d'application des honeypots le plus intéressant. Le rôle du honeypot dans un environnement de recherche est d'observer les méthodes de piratage et étudier le comportement des pirates. Ce type d'honeypot est plus difficile à mettre en place et requiert un suivi constant.

4. Selon les données capturées :

Décrit le type de données qu'un outil peut capturer du point de vue de l'attaque. Ses valeurs possibles (un outil peut avoir plusieurs valeurs assignées) sont:

- **Événement** : L'outil recueille des données sur les changements d'état,

-**Attaques :** L'outil collecte une activité malveillante (tentative de violation de la politique de sécurité),

- **Intrusions:** L'outil collecte une activité malveillante qui entraîne une défaillance de sécurité (craquage), c'est-à-dire un compromis ou une infection du système,

- **Aucun :** L'outil ne collecte pas d'événements, d'attaques ou d'intrusions.

5. Selon le contenu:

Décrit les mesures prises par un outil pour défendre / contraindre les activités malveillantes à se propager. Ses valeurs possibles (un outil peut avoir plusieurs valeurs assignées) sont:

- **Bloc:** l'activité malveillante est identifiée et bloquée (l'attaque n'atteint jamais la cible)
- **Désamorcer:** une activité malveillante est autorisée, mais elle est désamorcée (l'attaque atteint la cible, mais elle est manipulée de manière à ne pas réussir)
- **Ralentissement**: l'activité malveillante est ralentie,
- **Aucun:** aucune action n'est prise pour limiter l'activité malveillante.

6. Selon la distribution :

Décrit si le système honeypot semble être composé d'un seul système ou de plusieurs systèmes. Les valeurs possibles sont:

- **Distribué:** honeypot est ou semble être composé de plusieurs systèmes,
- **Stand-Alone:** honeypot est ou semble être un seul système.

7. Selon l'interface de communication :

Décrit les interfaces que l'on peut utiliser pour interagir directement avec l'honeypot. Les valeurs possibles sont:

- **Interface réseau:** l'outil peut être directement communiqué via une interface réseau,
- **Interface matérielle sans réseau:** l'outil peut être directement communiqué via une interface matérielle autre qu'une interface réseau (c'est-à-dire USB),
- **API du logiciel:** l'outil peut être communiqué via l'API du logiciel.

X. Avantages et inconvénients d'un Honeypot :

Etablissons dans cette partie les avantages et les inconvénients des honeypots : **[43]**

1. Intérêts :

- Ils permettent la détection rapide des attaques connues mais aussi de nouveaux types d'attaques.
- Les informations récoltées permettent de connaitre les intrus et leurs procédures d'action et ainsi mettre en place des stratégies de défense plus adéquates.
- Les informations collectées peuvent être utilisées contre les intrus (poursuites judiciaires …)
- Les honeypots ne sont pas une solution que l'on place pour résoudre un problème mais plutôt un outil à exploiter.
- Peuvent être utilisés pour la prévention, puisque le but des honeypots est de capturer un pirate et/ou de l'occuper un certain temps, temps pendant lequel il ne s'attaquera pas aux vrais systèmes de production.
- Un honeypot peut être plus spécifique. Différents types de honeypots peuvent être utilisés pour détecter différents types d'attaques.
- Ce sont des alliés très efficaces pour les IDS et peuvent être des solutions rapides à mettre en place.
- Le coût en terme de ressources matérielles est négligeable.
- Enfin, utiliser un honeypot au sein d'un réseau interne d'une entreprise se révèle être un outil redoutable pour la détection des actes de malveillance provenant de l'intérieur.

2. Inconvénients :

- Les honeypots doivent simuler des services et des systèmes utilisés par les vrais systèmes de production. Ainsi un intrus expérimenté peut facilement se rendre compte qu'il s'agit d'un piège.
- Si l'intrus arrive à avoir le control sur le honeypot, celui-ci peut être utilisé pour d'autres attaques.
- Le faux système peut être mis en place à côté des serveurs de production. Le honeypot doit être attractif afin de susciter l'intérêt, sinon, il sera ignoré et donc inutile.
- La charge de travail et les ressources nécessaires liée à l'exploitation des honeypots dépend directement du niveau d'interaction du honeypot, plus celle-ci est forte, plus la charge de travail sera importante.

XI. Avantages face à un simple IDS :

Un IDS ne détecte que les attaques connues et produit de nombreux faux positifs. Les pots de miel permettent de palier à ces deux problèmes. En effet, comme le trafic circulant à l'intérieur du honeypot n'a pas lieu d'être, on peut le considérer automatiquement comme suspect et ainsi supprimer les faux positifs. De plus comme nous laissons entrer le pirate, nous pouvons analyser l'attaque qu'il a conduit même si celle-ci était jusqu'alors inconnue. **[38]**

XII. Les Honeynets

Un Honeynet est différent de la plupart des honeypots, car il s'agit d'un réseau d'ordinateurs réels avec lequel les attaquants interagissent.

Conceptuellement, les honeynets sont très simples. Ils sont un réseau qui contient un ou plusieurs honeypots. Puisque les honeypots ne sont pas des systèmes de production, le honeynet lui-même n'a aucune activité de production, aucun service n'y est autorisé. En conséquence, toute interaction avec un honeynet implique une activité malveillante ou non autorisée. Toutes les connexions entrant dans un honeynet sont probablement une menace. Cela rend l'analyse de l'activité dans le honeynet très simple. Avec les technologies de sécurité traditionnelles, telles que les journaux de pare-feu ou les capteurs IDS, on doit passer en revue des Gigaoctets de données. On passe beaucoup de temps et d'efforts à examiner cette information, à essayer d'éliminer les faux positifs tout en identifiant les attaques ou les activités non autorisées. **[44] [38]**

XIII. Architecture d'un honeynet :

Pour déployer avec succès un honeynet, Nous devons déployer correctement l'architecture honeynet. La clé de l'architecture honeynet est ce que nous appelons **Honeywall**. Il s'agit d'un périphérique de passerelle qui sépare les honeypots du reste du monde. Tout trafic passant ou en provenance des honeypots doit passer par le honeywall. Cette passerelle est généralement un périphérique de transition de couche 2, ce qui signifie que le périphérique devrait être invisible pour quiconque interagissant avec les honeypots. Ci-dessous, nous voyons un diagramme de cette architecture. Le honeywall a 3 interfaces. Les 2 premières interfaces (Eth0 et eth1) séparent les honeypots du reste du réseau. La 3ème interface (eth2, facultative) permet une administration à distance.

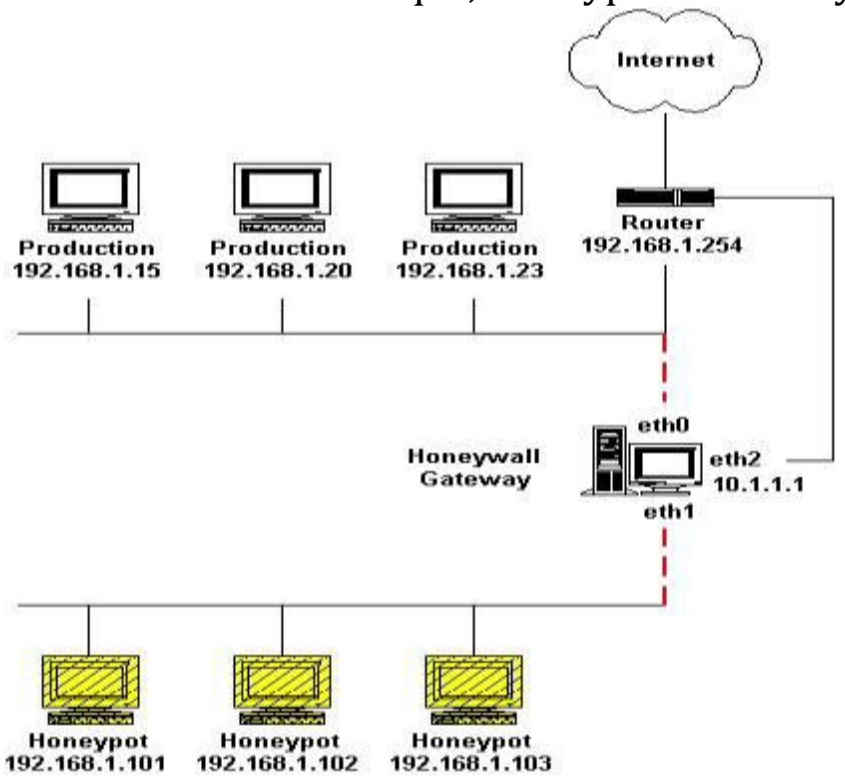

Figure 7: Architecture d'un honeynet. [44]

XIV. Considérations et conditions de déploiement :

Pour réussir le déploiement d'un honeypot ou d'un honeynet il faut répondre à certaines exigences. Ces dernières sont décrites selon les catégories suivantes: **[44] [40]**

1. Contrôle des données :

Comme déjà souligné, une chose importante à retenir lors du déploiement d'un honeypot est le facteur de risque qui y est associé. Un honeypot est conçu pour interagir avec des attaquants. Finalement, cela peut les amener à prendre une certaine forme de contrôle. Un attaquant peut réussir à obtenir des informations qui pourraient être utilisées pour des activités illégales telles que : compromettre d'autres systèmes, l'envoi de spams ou la propagation d'un ver…

En conséquence, le réseau où se situent les honeypots doit être étroitement contrôlé. Il est essentiel de surveiller et de contrôler le trafic entrant et sortant. Par exemple, il est judicieux de refuser les connexions sortantes, à l'exception de celles vers le site de l'initiateur et un ensemble d'hôtes prédéfinis tels que les serveurs DNS. Plus précisément, il est recommandé de bloquer au moins les connexions sortantes aux serveurs SMTP externes (port 25 / TCP) pour empêcher l'envoi de messages indésirables. Alternativement, des configurations plus élaborées peuvent

inclure la construction de faux SMTP, des proxies HTTP, des services DNS, etc., complétées par des enregistrements et des alertes, ce qui donne à l'équipe de sécurité autant de contrôle de l'environnement que possible, tout en le laissant apparaître de manière adéquate réaliste pour ne pas éveiller les soupçons.

D'une manière générale, les fichiers honeypots devraient être déployés dans un sous-réseau physiquement distinct, de sorte que le trafic réseau associé ne gêne pas le trafic légitime sur le réseau de production. D'autre part, il est tout à fait logique de mélanger les adresses IP utilisées dans un honeynet avec des adresses de réseaux de production. Une telle configuration nécessite une quantité importante de temps consacrée à la configuration des périphériques de routage, mais cet effort ponctuel serait compensé par le fait de rendre le Honeypots beaucoup plus difficile à distinguer des serveurs réels.

Les mécanismes de contrôle des données peuvent inclure : le déploiement des systèmes de détection / prévention des intrusions, des restrictions de bande passante et des pare-feu. Une combinaison de diverses techniques est toujours une bonne idée. Non seulement il élimine un seul point d'échec, mais il contribue également à protéger contre l'élimination d'un seul périphérique.

2. Capture de données :

Afin de comprendre comment les attaques sont menées et quelles techniques sont utilisées par les attaquants, il faut capturer toute l'activité associée au honeynet. Cela signifie que toutes les informations qui entrent ou sortent des honeypots doivent être enregistrées. Ceci, bien sûr, devrait être fait sans que l'attaquant le sache. Bien que les filtres offrent généralement leurs propres journaux, ils ne sont jamais complets. Par conséquent, il faut souligner que les outils réseau et système externes devraient être configurés pour enregistrer les données séparément.

Les données capturées doivent être stockées dans un emplacement différent de celui de l'honeypot lui-même, de sorte que si l'attaquant compromet un système honeypot, les données ne peuvent pas être modifiées ou détruites.

3. Sauvegarde de données :

En général, il est logique de stocker des données recueillies à partir d'un honeypot (ou d'un honeynet) en dehors de l'infrastructure qui est responsable de l'interaction directe avec un attaquant. Cela peut se faire de manière distribuée sur plusieurs serveurs ou dans des configurations plus simples, à un emplacement centralisé. La principale motivation est la

protection de l'intégrité des données (par exemple, pour empêcher les tentatives d'un attaquant de supprimer leurs traces). Lorsque tous les journaux et les fichiers binaires sont collectés et stockés en dehors des capteurs déployés, l'accès aux données est garanti indépendamment de ces capteurs. Les configurations exactes peuvent varier en fonction des besoins d'une organisation, de la quantité de données collectées, de l'infrastructure réseau, des ressources pouvant être engagées, etc. Elles peuvent être très individuelles.

4. L'analyse des données :

Il est essentiel d'avoir la possibilité d'analyser les données collectées, c'est-à-dire d'en extraire des informations précieuses. Cela peut inclure, par exemple, la recherche de nouveaux types d'attaques, de l'informatique légale post-intrusion ou de l'analyse de tendances à long terme. Les objectifs d'analyse peuvent donc avoir de sérieuses implications pour le processus de collecte et de stockage des données, décrits dans les sections précédentes. La plupart des filtres ne fournissent pas une classification complète des menaces découvertes, l'interprétation et l'analyse des données peuvent être des tâches très difficiles. Malheureusement, les outils analytiques manquent. Les administrateurs de Honeypots/Honeynets doivent choisir judicieusement ce qu'il faut collecter et analyser.

XV. Les Honeypots dans le Cloud Computing :

Le cloud computing est l'une des technologies qui sont en croissance rapide et est constamment menacé d'attaques. L'approche Honeypot est utilisée pour faire face à de telles attaques car elle attire non seulement le pirate pour attaquer le réseau, mais aussi alerte les administrateurs de réseau d'une possible intrusion en indiquant des informations sur l'attaquant. Les Honeypots peuvent être utilisés avec une autre forme de sécurité, comme un IDS pour augmenter leur efficacité. La mise en place d'un Honeynet semble alors être une bonne solution pour détecter ces nouvelles techniques que les attaquants utilisent pour détourner les Clouds et donc pour par la suite créer des systèmes de sécurité adéquats.

Conclusion:

Dans ce chapitre nous avons vu l'importance de la sécurité informatique face aux dangers des pirates informatiques et leurs mauvaises intentions. Nous avons aussi mit l'accent sur les Honeypots et Honeynets afin de mieux comprendre en quoi ils consistent et quels sont leurs buts, car c'est ce type de sécurité informatique que nous avons décidé de mettre en place sur notre environnement de Cloud Computing.

Chapitre III : Présentation des outils utilisés.

Introduction :

Dans les chapitres précédents nous avons donné l'état de l'art de tout ce que nous comptons mettre en place (Cloud et Honeypots) . Maintenant que nous nous sommes familiarisés avec ces concepts il est temps de les mettre en pratique.

Dans ce chapitre nous allons donc parler des outils choisis, leurs fonctionnements et comment les installer.

I. Openstack :

Comme l'a montré la comparaison des différents IAAS faite dans le premier chapitre, OpenStack apparait comme étant la meilleure solution pour le déploiement d'un cloud. C'est pour cela que nous avons opté pour son utilisation afin de mener à bien notre travail.

1. Definition :

OpenStack est un ensemble d'outils logiciels pour la construction et la gestion de plates-formes de cloud computing pour les nuages publics et privés. Soutenu par certaines des plus grandes entreprises dans le développement et l'hébergement de logiciels, ainsi que des milliers de membres de la communauté, beaucoup pensent qu'OpenStack est l'avenir du cloud computing. OpenStack est géré par la Fondation OpenStack, un organisme à but non lucratif qui supervise le développement et la construction communautaire autour du projet.

OpenStack fournit une solution Infrastructure-as-a-Service (IaaS) à travers une variété de services complémentaires. Chaque service offre une interface de programmation d'application (API) qui facilite cette intégration.

Il permet aux utilisateurs de déployer des machines virtuelles et d'autres instances qui gèrent différentes tâches pour la gestion d'un environnement cloud. Et surtout, OpenStack est un logiciel open source, ce qui signifie que toute personne qui choisit d'accéder au code source, effectue les modifications dont il a besoin et partage librement ces changements dans la communauté. Cela signifie également que OpenStack bénéficie à des milliers de développeurs dans le monde entier travaillant en tandem pour développer le meilleur produit possible. **[46]**

2. Les composants clés d'OpenStack :

OpenStack est composé de plusieurs pièces mobiles différentes. En raison de sa nature ouverte, n'importe qui peut ajouter des composants supplémentaires pour répondre à ses besoins. Mais les composants clés sont les suivants :

- **Keystone** fournit des services d'identité pour OpenStack. Il s'agit essentiellement d'une liste centrale de tous les utilisateurs du cloud OpenStack, mis en correspondance avec tous les services fournis par le cloud, qu'ils ont la permission d'utiliser.
- **Glance** fournit des services d'images à OpenStack. Dans ce cas, "images" se réfère à des des copies virtuelles de disques durs. Glance permet à ces images d'être utilisées comme modèles lors du déploiement de nouvelles instances de machines virtuelles.
- **Nova** est le principal moteur informatique derrière OpenStack. Il est utilisé pour déployer et gérer un grand nombre de machines virtuelles et d'autres instances pour gérer les tâches informatiques.
- **Neutron** fournit la capacité réseau pour OpenStack. Cela permet de s'assurer que chacun des composants d'un déploiement OpenStack peut communiquer rapidement et efficacement.
- **Horizon** est le tableau de bord derrière OpenStack. C'est la seule interface graphique pour OpenStack, donc c'est peut-être le premier composant que les utilisateurs voient réellement. Les développeurs peuvent accéder à tous les composants d'OpenStack individuellement via une interface de programmation d'application (API) mais le tableau de bord fournit aux administrateurs système un regard sur ce qui se passe dans le nuage et aide à le gérer au besoin. **[45] [46]**

3. Architecture minimale :

L'architecture minimale requiert au moins deux nœuds (hôtes) pour lancer une machine virtuelle ou une instance de base. Ces hôtes sont un nœud de contrôle et un nœud de calcul. Les services optionnels tels que le stockage de blocs et le stockage d'objets nécessitent des nœuds supplémentaires.

a. Controller (contrôleur) :

Le noeud du contrôleur exécute le service d'identité, le service d'image, les parties de gestion de Compute, la partie de gestion du réseau, les différents agents de mise en réseau et le tableau de bord. Il inclut également des services de support tels qu'une base de données SQL, une file d'attente de messages et NTP.

En option, le noeud du contrôleur exécute des portions des services de stockage de blocs, de stockage d'objets, d'orchestration et de télémétrie.

Le nœud du contrôleur nécessite au minimum deux interfaces réseau.

b. Compute (Calcule) :

Le nœud de calcul exécute la partie hyperviseur de Compute qui exploite les instances. Par défaut, Compute utilise l'hyperviseur KVM. Le nœud de calcul exécute également un agent de service de mise en réseau qui connecte des instances à des réseaux virtuels et fournit des services de pare-feu aux instances via des groupes de sécurité. Vous pouvez déployer plus d'un nœud de calcul. Chaque noeud nécessite au minimum deux interfaces réseau.

II. DevStack :

Le déploiement d'un test OpenStack ou d'une plate-forme de développement peut être une tâche très fastidieuse. Une installation traditionnelle d'une infrastructure Openstack nécessite de nombreux serveurs et est assez complexe. Cependant, il existe quelques méthodes qui peuvent rendre cette tâche beaucoup plus facile et possible avec l'accès à un seul serveur physique ou une machine virtuelle qui dispose de ressources suffisantes.

1. Définition :

DevStack est une série de scripts extensibles utilisés pour créer rapidement un environnement OpenStack complet basé sur les dernières versions. Il est utilisé interactivement comme un environnement de développement et constitue la base d'une grande partie des tests fonctionnels du projet OpenStack.

2. Installation d'OpenStack via DevStack:

Dans le cadre de ce projet, nous déployons une infrastructure Openstack Ocata à l'aide d'une seule machine virtuelle (dans notre cas, une machine virtuelle créée à l'aide du logiciel VMware avec pour OS Ubuntu).

1. Tout d'abord commençons par créer une machine virtuelle sous Ubuntu 16.04 (c'est la distribution recommandée pour cette version d'OpenStack) disposant d'assez de RAM (6 Go dans notre exemple), de mémoire disque (300 Go dans notre cas) et d'une interface réseau bridgée comme le montre la figure suivante :

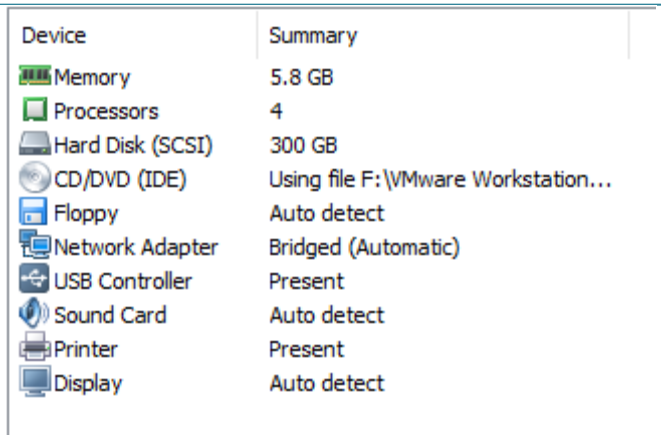

Figure 8: Ressources allouées pour la création de la machine virtuelle pour OpenStack.

2. Après, nous devons utiliser la commande git pour cloner devstack. Devstack fera tout le travail qui installe les composants d'OpenStack sur un seul serveur.

```
$ cd /

$ sudo git clone https://git.openstack.org/openstack-dev/devstack -b stable/ocata
```

3. Ensuite, nous devons copier l'exemple du fichier local.conf et définir un mot de passe qui sera utilisé pendant le déploiement automatisé.

```
$ cd devstack/

$ sudo cp samples/local.conf local.conf

$ sudo nano local.conf
```

Défiler vers le bas jusqu'à ce que nous voyons les variables de mot de passe. Il faut définir le mot de passe après « ADMIN_PASSWORD = » et modifier les trois autres à « $ADMIN_PASSWORD ». Cela fait que tout utilise le même mot de passe pendant l'installation.

```
ADMIN_PASSWORD=MotDePasseSeNotreChoix

MYSQL_PASSWORD=$ADMIN_PASSWORD

RABBIT_PASSWORD=$ADMIN_PASSWORD

SERVICE_PASSWORD=$ADMIN_PASSWORD
```

4. Ensuite, nous allons exécuter un script pour créer un nouvel utilisateur pour OpenStack, puis faire de ce nouvel utilisateur le propriétaire du dossier devstack.

```
$ sudo /devstack/tools/create-stack-user.sh

$ sudo chown -R stack:stack /devstack
```

5. Nous pouvons maintenant lancer l'installation.

```
$ sudo su stack

$ /devstack/stack.sh
```

Après un certain temps (environ 3h) nous avons obtenu un récapitulatif de l'installation comme le montre la figure suivante :

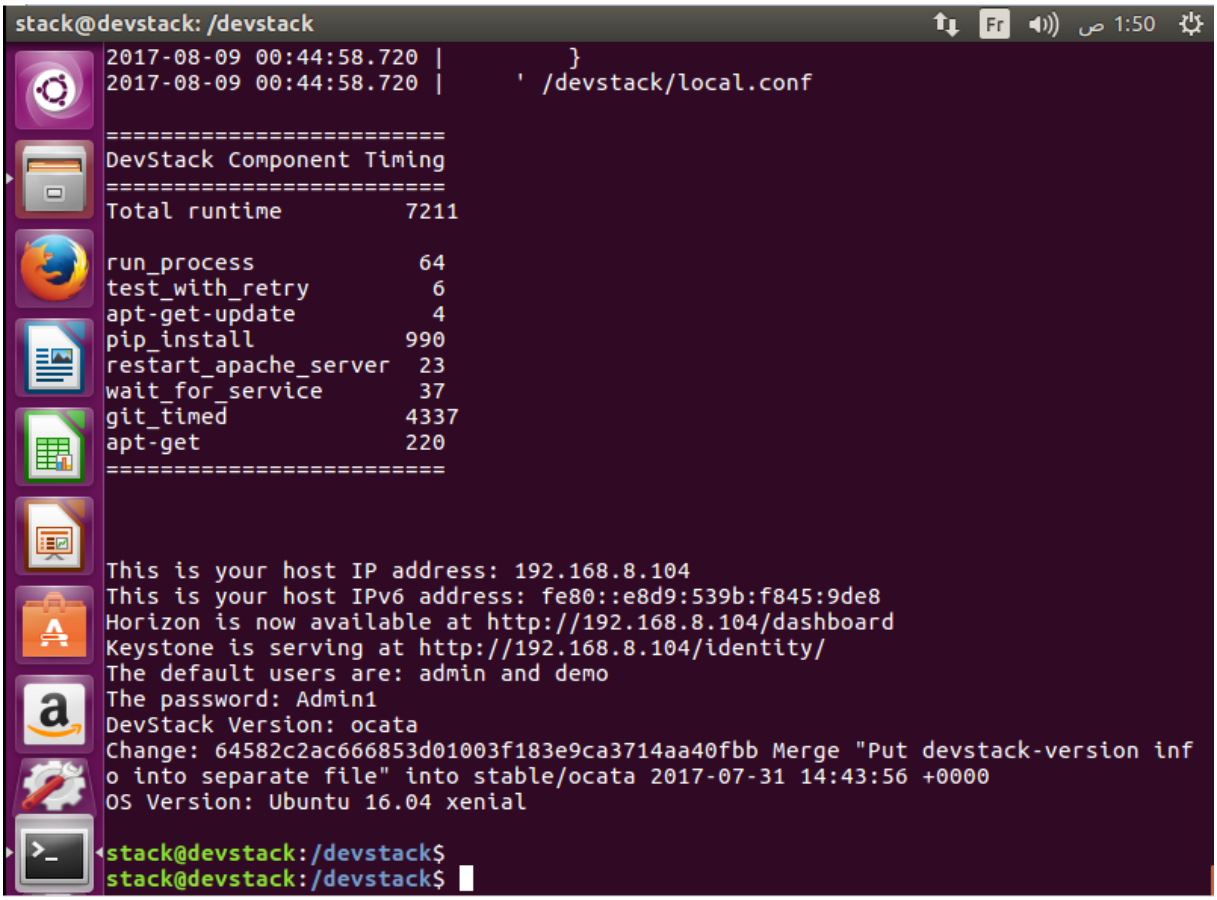

Figure 9: Screenshot de note VM à la fin de l'installation d'OpenStack via DevStack.

L'installation est maintenant finie.

6. Comme nous pouvons le voir, deux utilisateurs ont été créés; admin et démo. Notre mot de passe est le mot de passe que nous avons défini précédemment. Ce sont les noms d'utilisateur qu'il faut utiliserer pour se connecter au tableau de bord OpenStack Horizon. Il ne faut pas oublier l'adresse Web d'Horizon indiquée dans le terminal car c'est via cette adresse que notre cloud nous sera accessible.
7. Après avoir lancé un navigateur et mit l'adresse Horizon Dashboard dans la barre d'adresse. La notre est http://192.168.8.104/dashboard

   Une page de connexion apparait comme celle-ci :

Figure 10: Champs d'identification dans le dashboard.

Après la connexion, il est possible de commencer à utiliser le cloud (créer/supprimer des images, des instances …) et adapter le cloud à nos besoins. **[47]**

III. Dionaea :

Dionaea est un honeypot à faible interaction pour la capture de logiciels malveillants.Il est initialement développé sous l'édition 2009 de Google The Summer of Code (GSoC) du projet Honeynet. Dionaea vise à piéger les logiciels malveillants exploitant les vulnérabilités exposées par les services offerts sur un réseau et finalement obtenir une copie du logiciel malveillant.
Dionaea dispose d'une architecture modulaire, intégrant Python comme langage de script afin d'imiter les protocoles. Il est capable de détecter des codes shell en utilisant LibEmu. **[50]**

1. Protocoles à partir desquels Dionaea piège les Malwares :

-Server Message Block (SMB) : SMB est le protocole principal offert par Dionaea. SMB a une histoire connue de bugs exploitables à distance, et est une cible très populaire pour les vers.
-Hypertext Transfer Protocol (HTTP) : Dionaea prend en charge HTTP sur le port 80 ainsi que HTTPS. Un certificat SSL auto-signé est créé au démarrage pour HTTPS.
-Protocole de transfert de fichiers (FTP) : Dionaea fournit un serveur FTP de base sur le port 21. Il permet la création de répertoires, et le téléchargement de fichiers.
-Protocole de transfert de fichier trivial (TFTP) : Dionaea fournit un serveur TFTP sur le port 60 qui peut être utilisé pour diffuser des fichiers.
-Microsoft SQL Server (MSSQL) : Dionaea implémente le protocole Tabular Data Stream utilisé par Microsoft SQL Server. En écoutant TCP / 1433 et en permettant aux clients de se connecter, il peut décoder les requêtes exécutées sur la base de données.
-Voice over IP (VoIP) : le protocole VoIP utilisé dans Dionaea est le protocole initial de session (SIP). Ce module ne se connecte pas à un serveur / enregistreur VoIP externe; Il attend simplement les messages SIP entrants, enregistre toutes les données sous forme d'incidents et / ou de décharges de données binaires, et réagit en conséquence. **[48]**

2. Fonctionnement :

La machine supportant Dionaea doit avant tout réagir positivement à toute forme de contact venant de l'extérieur et logguer toute tentative. Pour cela, il existe un module et un script python portant tous les deux le même nom : **nfq**. Le module nfq gère la partie kernel tandis que le script nfq a pour tâche de simuler un service. Il ne suffit pas que la machine soit réceptive, il faut un minimum de sécurité, afin que la manipulation ne se retourne pas contre nous. Cette mission revient à **libemu**. Sa fonction première est d'émuler un processeur et de détecter, de profiler et parfois d'exécuter au sein d'un environnement distinct un shellcode. Cela permet, notamment dès le download du malware, de garder le contrôle sans faire de compromis sur les informations qui sont offertes. Cette copie du malware sera stockée en local pour dissection mais peut être également soumise à des tiers parties afin d'en tirer le maximum d'information. Enfin, il s'agit de logguer l'attaque pour en avoir une vision plus synthétique. **[50]**

Le fonctionnement est résumé dans la figure 11 suivante :

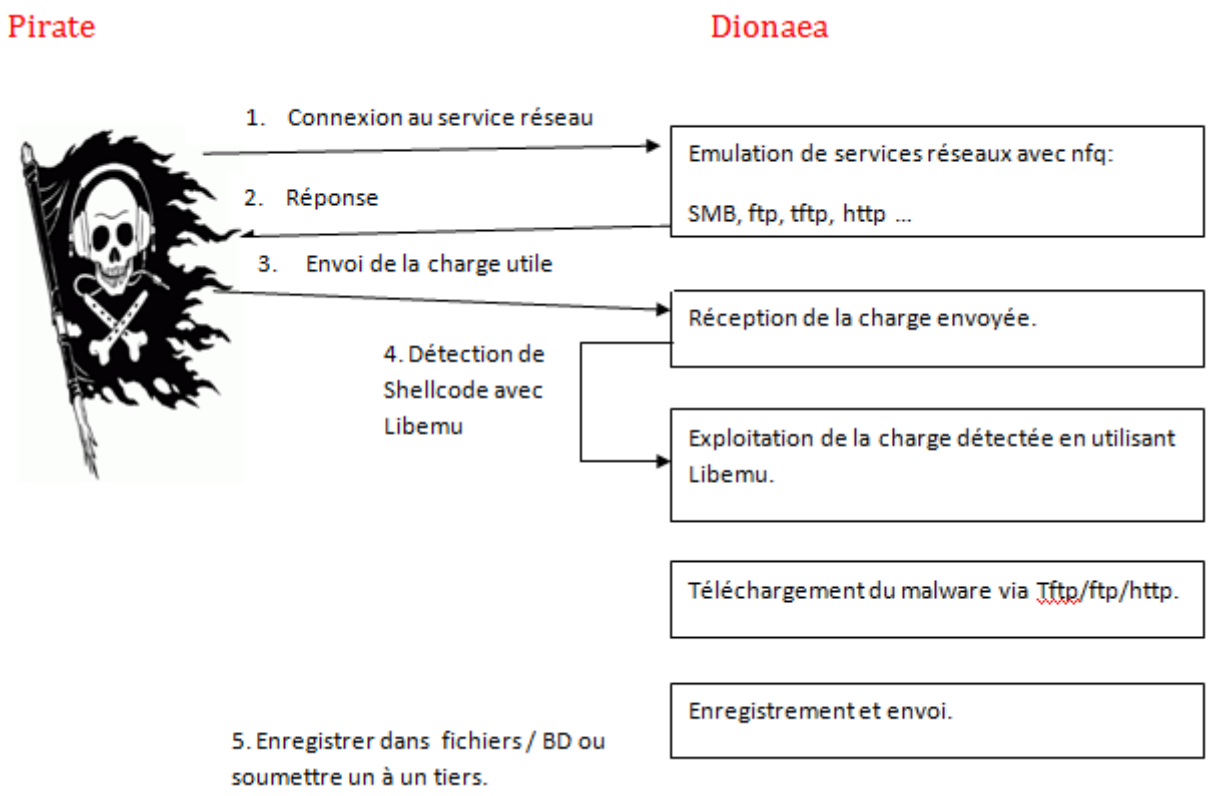

Figure 11: Schéma de fonctionnement de Dionaea.

- **Nfq :**

Le script python nfq est la contrepartie du module nfq. Alors que le module nfq interagit avec le noyau, le script nfq python prend en charge les étapes requises pour démarrer un nouveau service sur les ports. Nfq peut intercepter les connexions TCP entrantes pendant la poignée de main TCP donnant au honeypot la possibilité de fournir un service sur les ports qui ne sont pas servis par défaut. **[49]**

- **LibEmu :**

Dionaea utilise LibEmu pour détecter et évaluer les charges utiles envoyées par les attaquants afin d'obtenir une copie du logiciel malveillant.
LibEmu est utilisé pour détecter, mesurer et, si nécessaire, exécuter le shellcode. Les mesures / profils de Shellcode sont effectués en exécutant le shellcode dans LibEmu VM et en enregistrant les appels et les arguments de l'API. Ceci est suffisant pour le profilage de la plupart des shellcodes; mais pas pour les shellcodes multi-étages. En plus d'enregistrer les appels et les arguments de l'API, nous devons autoriser les shellcodes à prendre des mesures (par exemple, créer une connexion réseau)
Une fois que nous avons obtenu la charge utile et son profil, nous devons agir pour obtenir une copie du logiciel malveillant. **[51]**

3. Installation :

Pour pouvoir installer dionaea il nous faut d'abord une instance.

- **Création de l'instance du Honeypot :**

La dernière version d'Ubuntu sur laquelle tourne Dionaea est ubuntu 14.04. Comme nous sommes dans un environnement cloud il va falloir créer une instance sous cette version là d'ubuntu. La manière la plus simple est rapide de faire cela est de télécharger une image déjà existante depuis le site http://cloud-images.ubuntu.com/ et de créer une instance sur la base de cette image. Pour les images des différentes versions d'ubuntu le login de l'utilisateur est « ubuntu ». Il faudra ensuite définir un mot de passe grâce à un script que nous intégrons durant la création de l'instance. Exemple de script :

```
#cloud-config

password: mypassword

chpasswd: { expire: False }

ssh_pwauth: True
```

Une fois la création de l'instance finie, il faut vérifier bon fonctionnement de l'instance (login, réseau, etc…).
Pour voir la création d'instances plus en détails consulter l'annexe.

- **Installation du service Dionaea :**
  Voici les étapes à suivre pour l'installation du Honeypot Dionaea :

1. Il faut tout d'abord mettre à jour tous les packages pour obtenir les dernières mises à jour de sécurité.

```
sudo apt-get update
sudo apt-get dist-upgrade
```

2. Ensuite, installer les outils pour gérer facilement les ressources PPA (Personal Package Archives) :

```
sudo apt-get install software-properties-common
```

3. Une fois les outils nécessaires installés, il faut ajouter le PPA et mettre à jour le cache du package.

```
sudo add-apt-repository ppa:honeynet/nightly
sudo apt-get update
```

4. S'il n'y a eu aucune erreur, l'installation du paquet dionaea se fait via la commande :

```
sudo apt-get install dionaea
```

5. Enfin, le service dionaea peut être démarré en utilisant la commande suivante :

```
sudo service dionaea start
```

C'est ce que montre la figure 12 suivante :

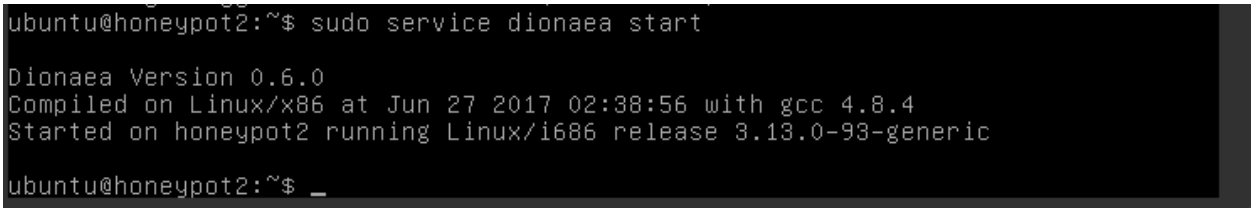

Figure 12: Lancement de Dionaea.

Les logs files se trouvent à l'emplacement : /opt/dionaea/var/dionaea/

IV. P0f :

Le nom est l'acronyme de « passive operating system fingerprinting » qui se traduit en : empreinte passive du système d'exploitation. P0f est différent des autres outils d'identification du système d'exploitation car il n'envoie aucun paquet à la cible. Il extrait plutôt les paquets et les examine pour déterminer le système d'exploitation qui les a envoyés. Il est toutefois important de noter que la fiabilité de la reconnaissance passive est inférieure à celle de la reconnaissance active. **[54] [55]**

Afin que la collecte de notre Honeypot soit plus complète nous allons y ajouter p0f.

1. Capacités de p0f :

Certaines des capacités de p0f comprennent:

- Identification rapide du système d'exploitation et du logiciel sur les deux extrémités d'une connexion TCP.

-Mesure du temps de disponibilité du système et de la connexion réseau, de la distance (y compris la topologie derrière les filtres NAT ou paquets).

- L'outil peut être utilisé au premier plan ou en démon, et offre une API simple en temps réel pour les personnes qui souhaitent obtenir des informations supplémentaires sur les acteurs interagissant avec leurs machines.

- Les utilisations communes pour p0f comprennent la détection des tests de pénétration; surveillance systématique du réseau; détection d'interconnexions réseau non autorisées dans les environnements d'entreprise. **[52]**

2. Fonctionnement de p0f:

Chaque système d'exploitation implémente la pile TCP / IP de manière légèrement différente et, par conséquent, les paquets TCP envoyés via ce système d'exploitation ont des attributs légèrement différents. Si nous pouvons extraire un paquet, l'examiner et chercher ces attributs, nous pouvons les comparer à une base de données d'attributs et déterminer avec un degré de précision relativement élevé quel système d'exploitation a envoyé le paquet. **[53]**

L'image ci-dessous est celle d'un en-tête TCP / IP. L'en-tête IP est ombré en jaune et l'en-tête TCP est en blanc. Quatre champs de cet en-tête, qui sont essentiels, sont entourés pour déterminer le système d'exploitation qui a généré le paquet. Ces champs sont montrés dans la figure 13.

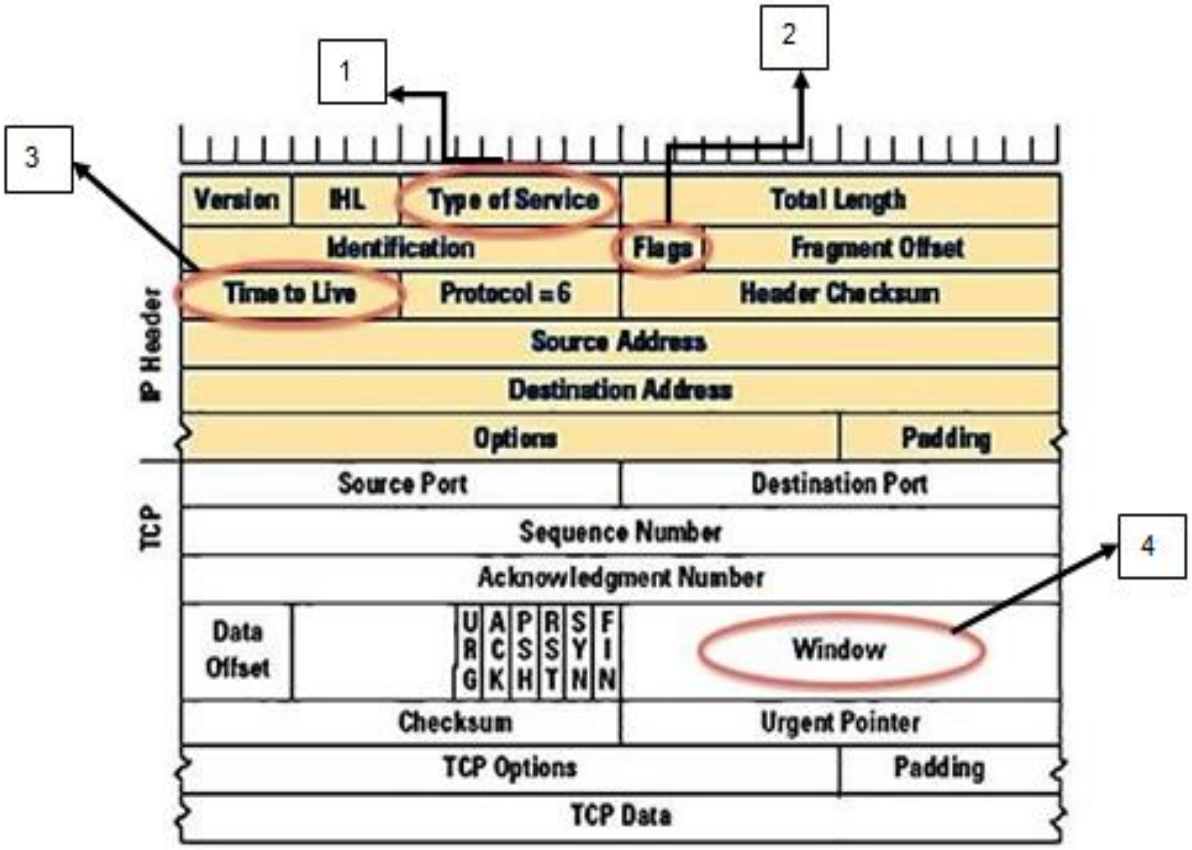

Figure 13: Entête d'une trame TCP/IP. [55]

- **Premier: type de service (TOS) :**

Le champ service « Type Of Service » est codé sur 8 bits (figure 14), il permet la gestion d'une qualité de service traitée directement en couche 3 du modèle OSI.

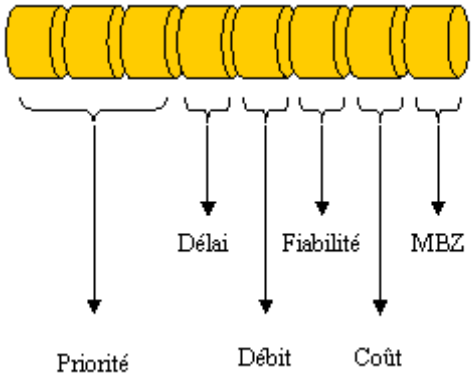

Figure 14: Champ service sur 8bits. [56]

- **Deuxième: drapeaux (DF) de l'en-tête IP :**

Le champ Flags est codé sur 3 bits et indique l'état de la fragmentation. Voici le détail des différents bits constituant ce champ.

- **Reserved :** Le premier bit est réservé et positionné à 0.
- **DF :** Appelé DF « Don't Fragment », le second bit permet d'indiqué si la fragmentation est autorisée. Si un Datagramme devant être fragmenté possède le flag DF à 1, alors, il sera alors détruit.
- **MF :** Appelé MF « More Fragments », le troisième bit indique s'il est à 1 que le fragment n'est pas le dernier. **[56] [55]**

- **Troisièmement: le temps de vivre (TTL) :**

Le champ TTL (Time To Live) est codé sur 8 bits et indique la durée de vie maximale du paquet. Chaque système d'exploitation définit le TTL au nombre maximal de sauts que le paquet peut traverser avant qu'il ne soit détruit. Généralement, les systèmes Windows ont cet ensemble à 32 tandis que les systèmes Linux ont le TTL réglé sur 64. **[56] [55]**

- **Quatrième: le champ Windows :**

Le champ Fenêtre « Windows » est codé sur 16 bits et correspond au nombre d'octets à partir de la position marquée dans l'accusé de réception que le récepteur est capable de recevoir. Le destinataire ne doit donc pas envoyer les paquets après Numéro de séquence + Window. C'est probablement le champ le plus important pour p0f car c'est celui qui vari le plus parmi les systèmes d'exploitations. Si nous pouvons trouver cette valeur, nous avons environ 70 à 80% de chances de déterminer le système d'exploitation qui a envoyé le paquet. **[57] [55]**

3. Installation et lancement de p0f:

L'installation de p0f se fait via une seule commande:

```
$ sudo apt-get install p0f
```

L'installation ne prend que quelques secondes.
Pour lancer p0f la commande suivante est utilisée :

```
$ p0f –i <nom de l'interface réseau ex : eth0> -vt
```

L'option –i indique qu'il s'agit de cet appareil,
-v affiche les flags de mascarades
-t définit le seuil de détection de mascarade (1-200)

V. Honeywall :

Le Honeywall est l'élément central de notre Honeynet, c'est lui qui va assurer le contrôle de données tandis que le Honeypot assure la collecte de données. Un Honeywall est un peu un firewall qui marche à l'envers, c'est à dire, il ne fait pas de restriction sur le flux entrant mais plutôt sur le flux sortant. Tout simplement car tout ce qui vient de l'attaquant nous intéresse tandis que nous ne voulons pas lui laisser une trop grosse marge de manœuvres.

Mise en place :

1. Il faut d'abord créer une instance. Exactement de la même manière que celle où nous avons installé le honeypot (voir annexe).
2. Maintenant que nous avons créé l'instance du Honeywall, il faut la transformer en passerelle entre l'instance du Honeypot et le router. Pour cela il faut bridger notre interface réseau, vu que l'instance n'en possède qu'une seule, et autoriser la retransmission de paquets.
3. Une fois que l'instance fait son travail de routage, il faut maintenant établir les règles du firewall que nous voulons établir grâce à ufw.

   Dionaea simule des services grâce à nfq, qu'il détecte et simule le déroulement d'un malware sur une VM grâce à Libemu. Donc le pirate n'est jamais en connexion avec les vraies ressources de notre machine. C'est pour cela que nous avons décidé de sécuriser le moins possible notre Honeypot (moins il y a de sécurité, plus il y a d'attaques à détecter). Nous avons donc autorisé les flux entrants et sortants tout en bloquant les connexions sortantes aux serveurs SMTP, comme cité dans le chapitre précédent, afin d'éviter l'envoie de messages indésirables.

   Enfin nous avons configuré l'option de logging dans le fichier ufw.conf à « full » afin d'enregistrer toutes les connexions sans restrictions.

VI. Nmap :

Nmap ("Network Mapper") est un utilitaire gratuit et open source pour la découverte de réseau et l'audit de sécurité. De nombreux systèmes et administrateurs de réseau le trouvent également utile pour des tâches telles que l'inventaire réseau, la gestion des calendriers de mise à niveau des services et la surveillance du temps de disponibilité de l'hôte ou du service. Nmap utilise des paquets IP bruts pour déterminer quels hôtes sont disponibles sur le réseau, quels sont les services (nom et version de l'application) que ces hôtes proposent, sous quels systèmes d'exploitation (et versions OS) ils fonctionnent, quel type de filtre / firewalls de paquets sont utilisés et des dizaines d'autres caractéristiques. Il a été conçu pour numériser rapidement de gros réseaux, mais fonctionne tout aussi bien avec les hôtes individuels. Nmap fonctionne avec tous les principaux systèmes d'exploitation informatiques, et des paquetages binaires officiels sont disponibles pour Linux, Windows et Mac OS X. **[58]**

1. Caractéristiques ne nmap :

Nmap est :

- **Flexible:** prend en charge des douzaines de techniques avancées pour la cartographie de réseaux remplis de filtres IP, pare-feux, routeurs et autres obstacles. Cela comprend de nombreux mécanismes de balayage de port (TCP et UDP), la détection d'OS, la détection de version, les balayages de ping et plus encore.
- **Puissant:** Nmap a été utilisé pour analyser d'énormes réseaux de centaines de milliers de machines.
- **Portable:** la plupart des systèmes d'exploitation sont pris en charge, y compris Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga et plus encore.
- **Facile:** bien que Nmap offre un ensemble complet de fonctionnalités avancées pour les utilisateurs puissants, sa prise en main est relativement simple.
- **Gratuit:** les objectifs principaux du projet Nmap sont d'aider à rendre l'Internet un peu plus sécurisé et de fournir aux administrateurs / auditeurs / whitehat informatiques un outil avancé pour explorer leurs réseaux. Nmap est disponible pour téléchargement gratuit, et contient également un code source complet modifiable.
- **Bien documenté:** Des efforts considérables ont été déployés dans des pages manuelles complètes et à jour, des livres blancs, des tutoriels et même un livre entier.
- **Prise en charge:** si Nmap est livré sans garantie, il est bien pris en charge par une communauté dynamique de développeurs et d'utilisateurs. La plupart de cette interaction se produit sur les listes de diffusion Nmap. La plupart des rapports de bogues et des questions doivent être envoyés à la liste nmap-dev.
- **Acclamé:** Nmap a remporté de nombreux prix, dont le «Produit de sécurité de l'information de l'année» par Linux Journal, Info World et Codetalker Digest. Il a été présenté dans des centaines d'articles de magazines, de plusieurs films, de dizaines de livres et d'une série de bandes dessinées.
- **Populaire:** des milliers de personnes téléchargent Nmap tous les jours. Il est également inclus avec de nombreux systèmes d'exploitation (Redhat Linux, Debian Linux, Gentoo…). Il figure parmi les dix premiers (sur 30 000) programmes au dépôt FreshMeat.Net. **[58]**

2. Fonctionnement :

Pour scanner les ports d'un ordinateur distant, Nmap utilise diverses techniques d'analyse qui s'appuient sur des protocoles tels que TCP, IP, UDP ou ICMP. De même, il se fonde sur les réponses qu'il obtient à des requêtes particulières pour obtenir une empreinte de la pile IP, souvent propre au système qui l'utilise. C'est par cette méthode qu'il peut reconnaitre la version d'un système d'exploitation ainsi que la version des services (aussi appelés daemons) en écoute. **[59]**

La figure 15 suivante, montre comment fait nmap pour détecter si un port X est ouvert ou fermé.

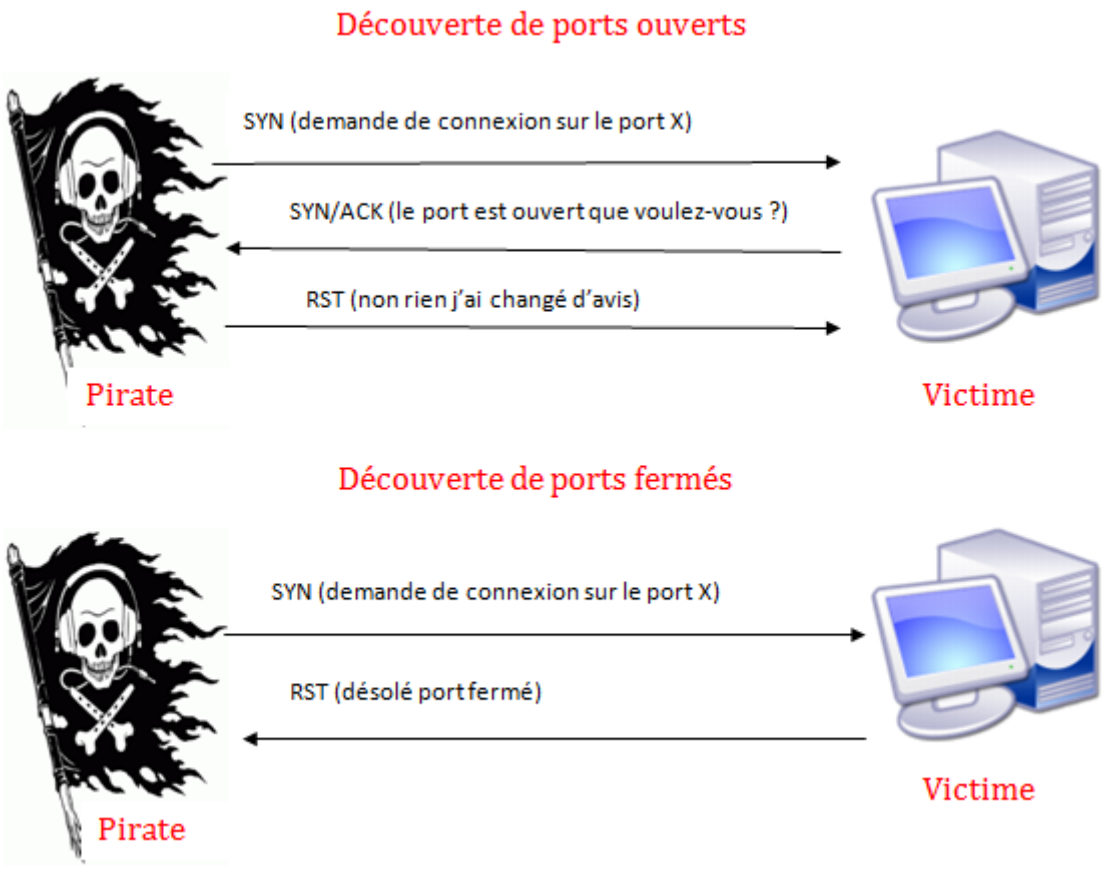

Figure 15: Détection de ports ouverts/fermés avec nmap.

3. Ouvrir Nmap :

Afin de ne pas perdre de temps à installer des outils d'attaques informatiques alors que ce n'est pas le but premier de notre travail, nous avons décidé d'utiliser une machine virtuelle sous Kali linux. Cette dernière est déjà dotée de toutes sortes d'outils de piratages.

Pour trouver nmap sur kali, il faut aller sur : **Applications** -> **Kali Linux** -> **Récupération d'information** -> **Détection de Service** -> **nmap.** Comme le montre la figure 16.

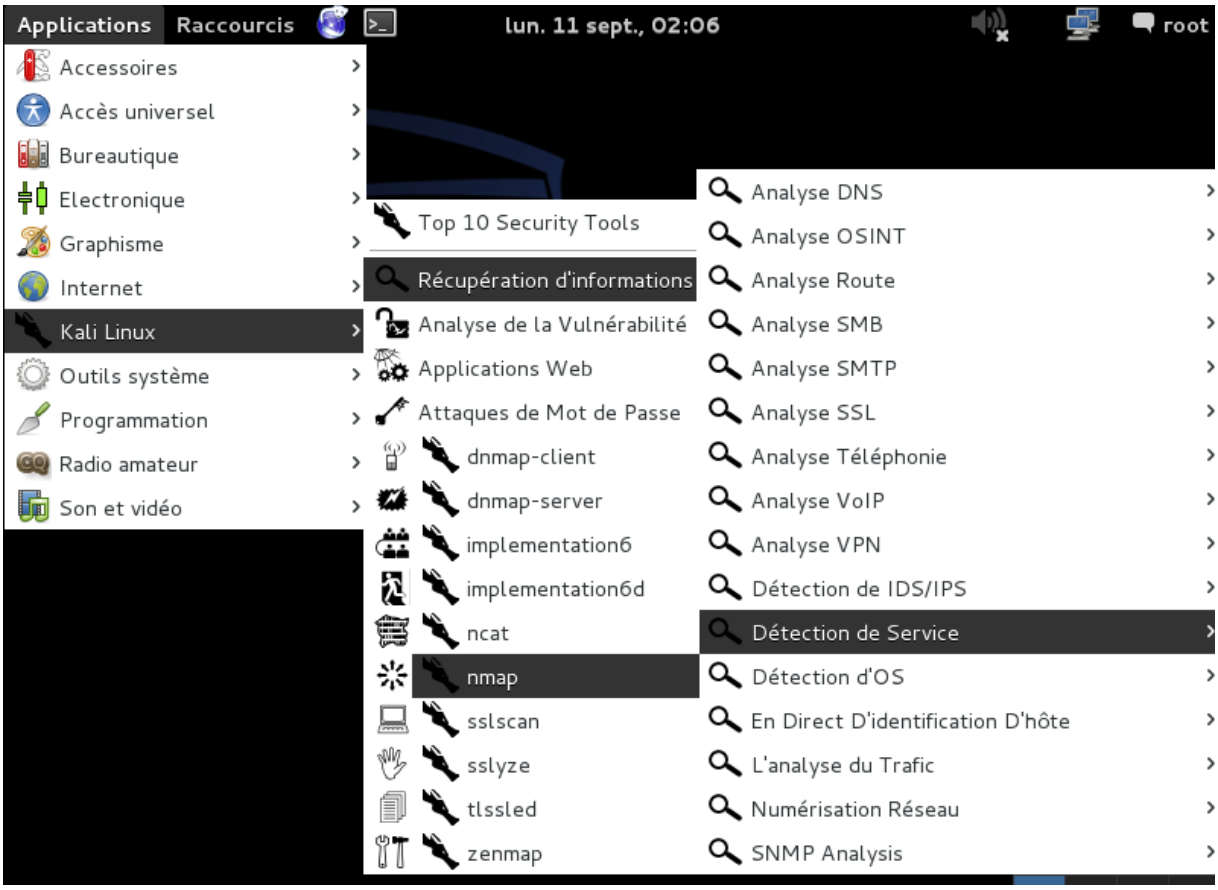

Figure 16: Ouverture de nmap sur une machine Kali Linux.

VII. Netcat :

Netcat, également abrégé en nc, a été créé pour être un outil d'analyse de réseau. En tant que tel, il peut être utilisé pour ouvrir les connexions TCP et UDP entre deux machines sur n'importe quel port souhaité et c'est cette option qui nous intéresse. Il est conçu pour être incorporé aisément dans un large panel d'applications. En raison de sa polyvalence, netcat est aussi appelé le « couteau suisse du TCP/IP ». Il existe sur plusieurs systèmes d'exploitation et s'utilise en ligne de commande. Il peut également être utilisé comme un outil de balayage de port, similaire à nmap.

Netcat peux fonctionner dans l'un des deux modes de base. En tant que client, Netcat opère avec l'intention de lancer une connexion à un autre ordinateur (ou au même ordinateur). Inversement, le même Netcat fonctionne en mode serveur ou auditeur lorsque des paramètres spécifiques sont transmis à l'utilitaire. **[60]**

1. Utilisation de netcat:

Netcat peut être utilisé à diverses fins parmi lesquelles :

- Se connecter à un système distant ;
- Exploiter la bannière d'un server web pour détecter son OS ;
- Ecouter les connexions sur un port en particulier ;
- Créer une porte dérobée ;
- Copier des fichiers (les exfiltrer) de la cible ;
- Faire du shell binding et du shell reverse…

Exemple :

Il existe deux types populaires de shell: bind et reverse. Un shell binding est le genre qui ouvre un nouveau service sur la machine cible, et exige que l'attaquant se connecte à lui pour obtenir une session. Un shell inversé (également connu sous le nom de "connect-back") est l'inverse: il faut que l'attaquant configure un auditeur d'abord sur sa boîte, la machine cible agit en tant que client se connectant à cet auditeur, puis enfin l'attaquant reçoit le shell.

- En utilisant un shell bind:Comme montré dans la figure 17, on lie un shell à un port TCP (étape 1). On peut alors se connecter et exécuter la commande (étape 2):

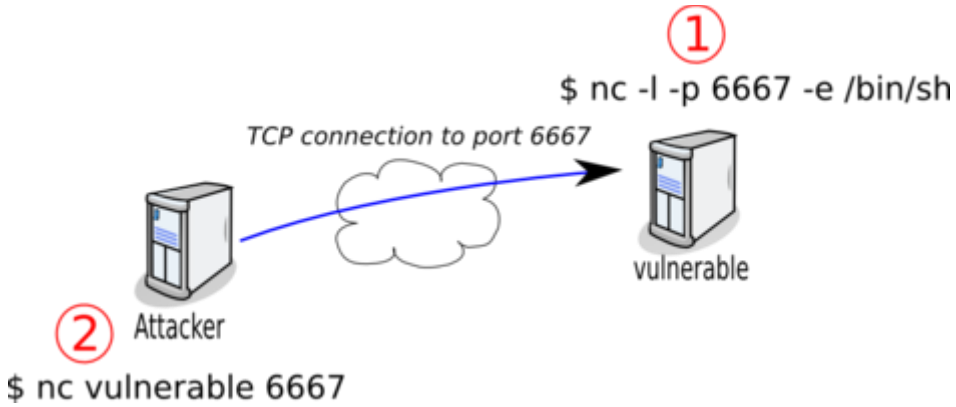

Figure 17: Exemple de shell binding avec Netcat. [61]

- A l'aide d'un shell inversé: Comme le montre la figure 18, on lie un port sur notre système local (étape 1) tout en permettant à la victime de se connecter à ce port. Ensuite on redirige les entrées/sorties vers un shell. Vous pourrez ensuite exécuter des commandes :

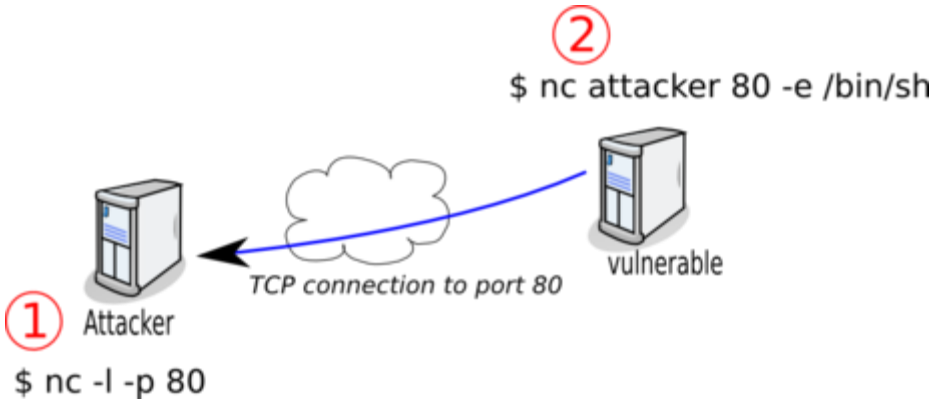

Figure 11: Exemple de shell reverse avec Netcat. [61]

2. Ouvrir Netcat :

Ici aussi nous continuons à utiliser la même machine virtuelle Kali Linux.

Une fois que nous avons démarré notre système Kali et ouvert un terminal, nous pouvons utiliser netcat en tapant la commande :

```
# nc –h
```

Cette commande nous affiche le résultat montré sur la figure 18, qui n'est autre que l'aide de netcat.

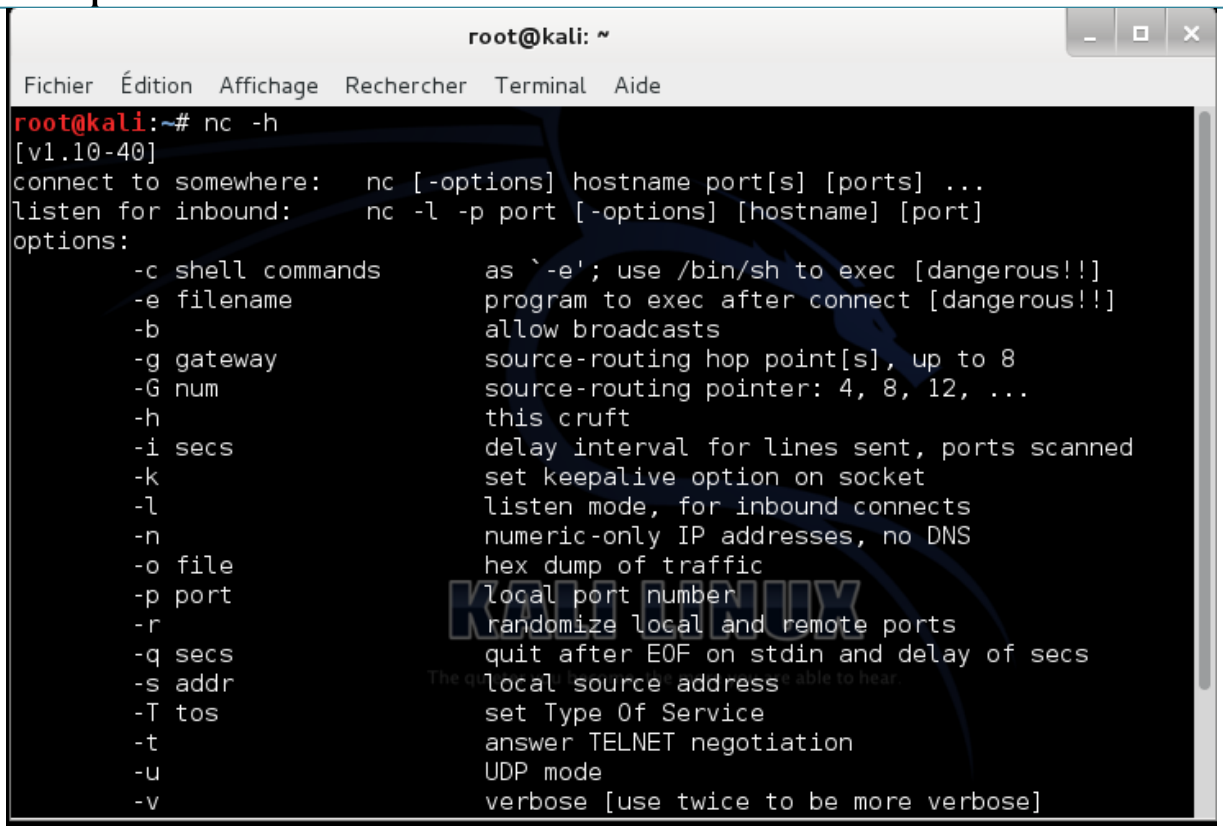

Figure 18: Ouverture de Netcat sur une machine Kali Linux.

Conclusion:

Nos outils sont maintenant tous installés, le Cloud est mis en place et est doté d'un Honeynet.

La machine avec laquelle nous allons simuler des attaques est aussi prête.

Maintenant il ne nous reste plus qu'à tester tout ça et à décortiquer les résultats obtenus.

Chapitre IV : Tests et résultats.

Introduction:

Notre environnement est maintenant en place. Il est temps de simuler des attaques sur notre Honeynet. Après ça nous allons montrer les résultats obtenus sur le Honeypot et décrypter un peu tout ça.

I. Topologie de notre réseau :

La figure 19 est un récapitulatif de notre topologie.

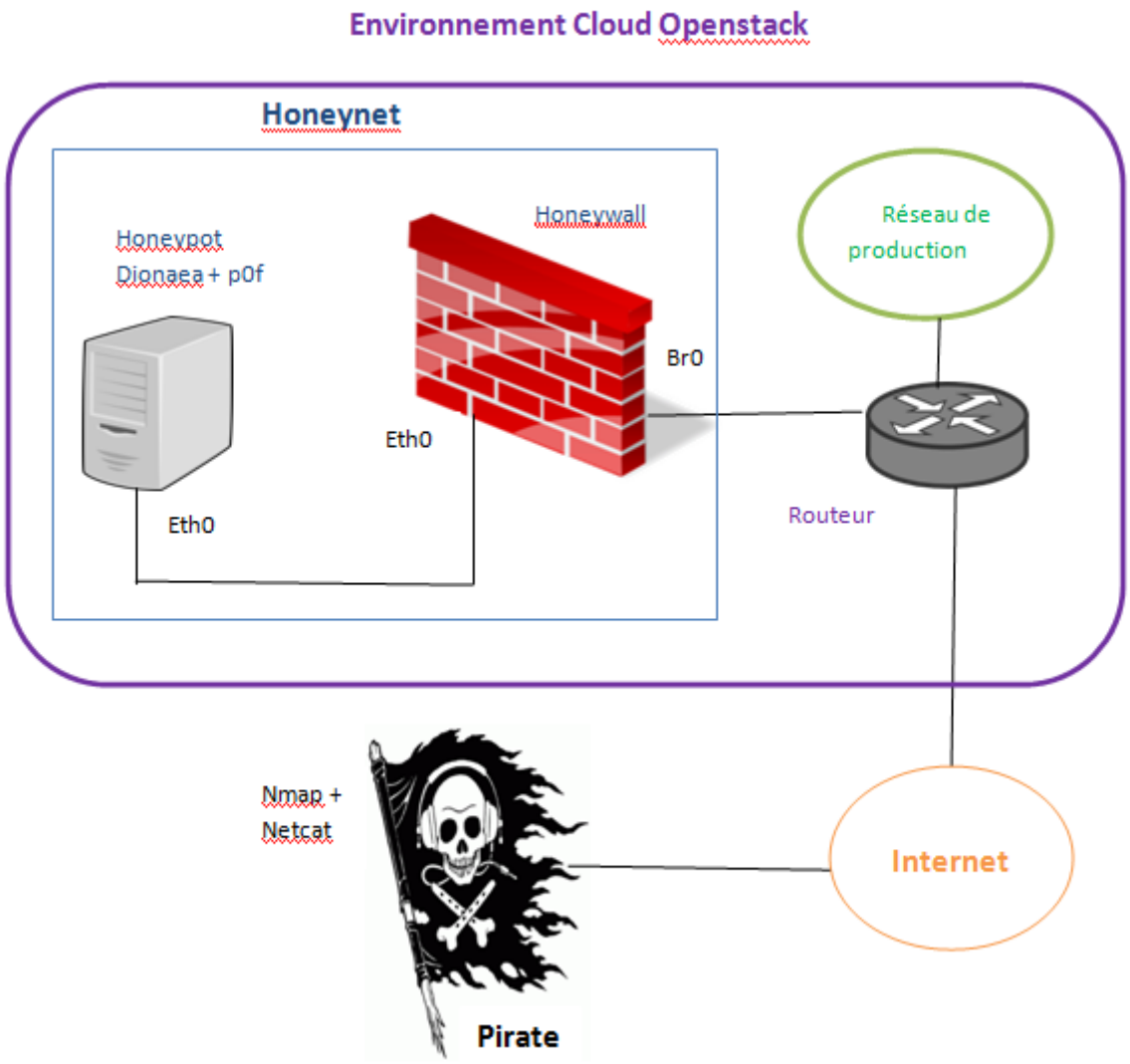

Figure 19: Topologie de notre réseau.

Tout d'abord le Honeywall enregistre toutes les connexions entrantes et sortantes de l'attaquant. Ceci est important car c'est notre première indication de ce que l'attaquant essaie de faire.

Ensuite le Honeypot enregistre toutes les activités de l'attaquant dans un log file. En collaboration avec p0f, dionaea nous donnera toutes les informations nécessaires sur l'attaquant. Le fichier log de Dionaea devrait être transféré sur un tiers serveur afin d'éviter que l'attaquant ne le supprime s'il se rend compte du piège dont lequel il se trouve. Cependant ne disposant pas des ressources nécessaires afin de créer plus de deux instances, nous avons laissé le fichier log sur l'instance du Honeypot.

II. Scan de ports avec Nmap :

A l'aide de nmap nous allons scanner les ports du honeypots. La figure 20 suivante montre le résultat du scan :

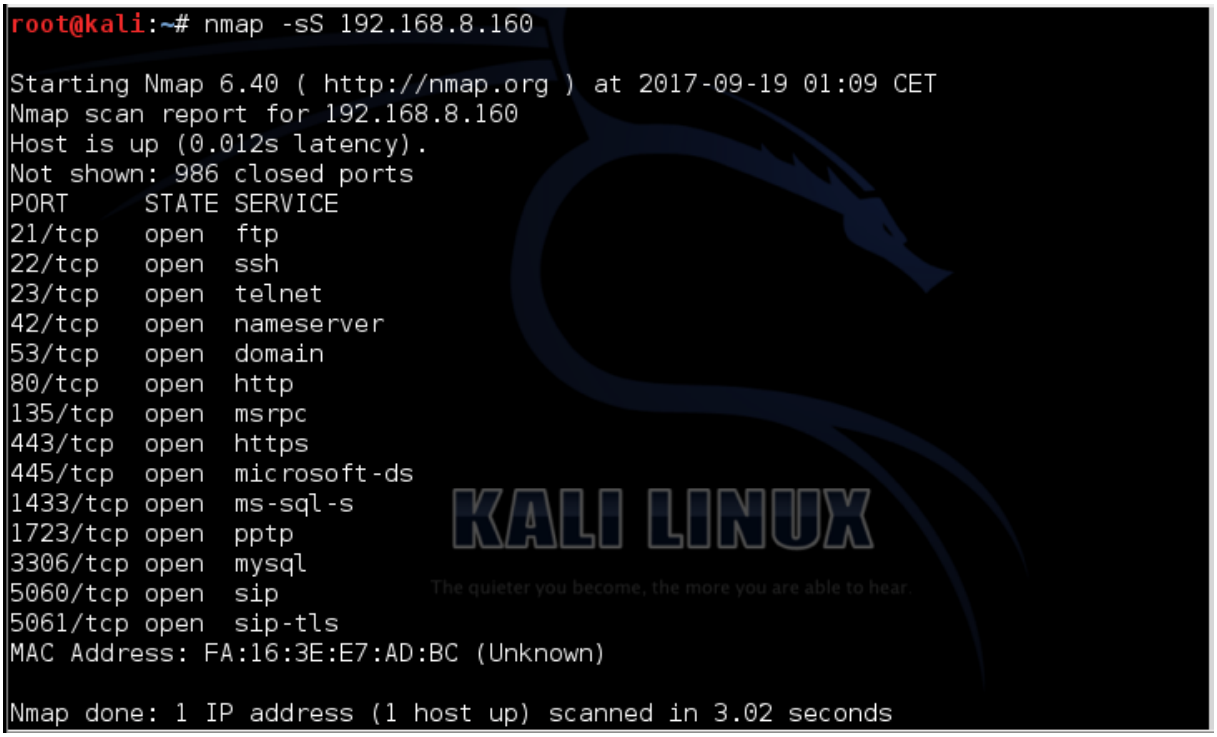

Figure 20: Scan nmap du Honeypot.

La première chose que l'on remarque sur ce scan est que les ports qui sont détectés comme étant ouverts sont les ports à partir des quels Dionaea piège les malware. Ceci ce fait grâce au script nfq qui simule ces services. Donc ces ports ne sont en réalité pas ouverts mais Dionaea fait croire à l'intrus qu'ils le sont pour l'encourager à nous attaquer.

1. Résultat de p0f :

Jetons maintenant un œil sur ce qu'à capter notre détecteur d'OS p0f dans la figure 21:

```
:?]
  -> 10.0.0.5:25735 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:52:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1192 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:38:0:44:M1460:.:?
:?]
  -> 10.0.0.5:3268 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:58:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1026 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:43:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1054 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:49:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1082 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:44:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1022 (link: ethernet/modem)
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:39:0:44:M1460:.:?
:?]
  -> 10.0.0.5:11967 (link: ethernet/modem)
```

Figure 21: Résultats de p0f suite au scan de ports.

Pour commencer, il y a autant de lignes que de ports scannés. Chaque scan de port a été détecté et perçu comme une attaque. Donc pour chaque tentative de connexion à un port, p0f isole le paquet TCP ressue, en extrait les attributs qu'il analyse par la suite pour en déduire le système d'exploitation de l'attaquant ainsi que certaines informations le concernant.

Regardons de plus près l'une des lignes et décryptons-la :

```
<Fri Sep 15 23:31:44 2017> 192.168.8.107:42837 - UNKNOWN [1024:58:0:44:M1460:.:?
:?]
  -> 10.0.0.5:1026 (link: ethernet/modem)
```

-P0f commence par indiquer la date et l'heure de l'attaque, ici c'est le vendredi 15 septembre 2017 à 23 :31 :44.

-Ensuite p0f indique l'adresse IP de l'attaquant : 192.168.8.107 via son port 42837.

-La longue suite de caractère [1024 :58 :0 :44 :M1460 :. :?:?] correspond à l'empreinte de l'OS détecté. Afin de savoir à quel système correspond ce code il faut regarder dans le fichier /etc/p0f/p0f.fp. Cette détection n'est pas toujours précise à cause du caractère passif de p0f.

-Après ça p0f pointe l'adresse IP de notre machine attaquée dans le réseau privé qui est donc 10.0.0.5 suivi du port qui a été sollicité ici c'est le 1026.

-Enfin p0f indique le type de connexion utilisé dans notre cas Ethernet/modem.

2. Fichier Log de Dionaea :

Passons maintenant à notre Honeypot et voyons ce qu'il a retenu du scan de port dans la figure 22:

Figure 22: Fichier log de Dionaea après le scan de ports.

Il est clair que ce qui se trouve entre crochets représente la date et l'heure exemple : [15092017 23 :32 :20] pour le 15 septembre 2017 à 23h32min20sec.

Dionaea utilise un système de communication interne qui s'appelle « incident ». Un incident a une origine, qui est une chaîne, un chemin d'accès et des propriétés, qui peuvent être des entiers, des chaînes ou un pointeur vers une connexion. Les incidents transmettent les informations requises aux gestionnaires d'incidents (ihandler).

Le python module permet d'utiliser l'interpréteur python dans Dionaea.

Le module pcap utilise la bibliothèque libpcap pour détecter les tentatives de connexion rejetées, alors même si nous n'acceptons pas une connexion, nous pouvons avoir l'information que quelqu'un souhaitait se connecter.

Log_sqlite c'est ce que fait le script logsql python, c'est un ihandler et écrit les incidents dans une base de données sqlite, l'un des avantages de cette connexion est la possibilité de regrouper les incidents en fonction de l'attaque initiale lors de la récupération des données de la base de données.

Ici chaque port scanné engendre plusieurs lignes de résultats. Cela commence avec pcap qui détecte la tentative de connexion en incluant le port ciblé sur notre IP ainsi que l'adresse IP de l'attaquant et du port qu'il a utilisé. Ceci provoque des incidents qui sont enregistrés la base de données sqlite.

III. Attaque netcat :

Nous avons utilisé netcat afin de nous connecter au server à distance via le port 80.

Les commandes tapés et les résultats obtenus sont dans la figure 23.

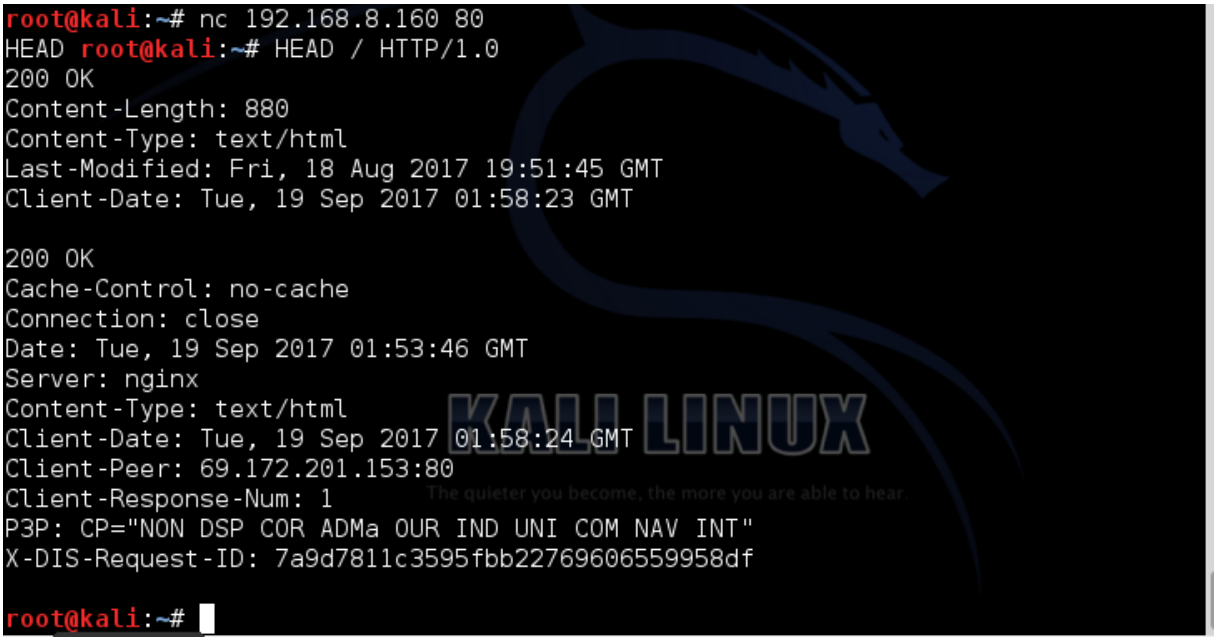

Figure 23: Connexion à distance sur le port 80 via netcat.

La commande nc 192.168.8.160 80 nous donne une connexion TCP, au serveur Web (port 80). Maintenant, tout ce que nous taperons sur le clavier sera envoyé directement au serveur web lorsque nous cliquons sur Entr.

Une fois que nous avons une connexion, nous pouvons saisir la bannière du serveur Web en tapant: HEAD / HTTP / 1.0

Dionaea fait croire alors que la machine dispose d'un serveur Web et répond en nous indiquant avec quel logiciel il fonctionne soit disant. Dans ce cas, nous pouvons voir que le serveur Web exécute nginx qui est un logiciel très répondu.

1. Résultat p0f :

Voyons si p0f arrive à détecter cette connexion dans la figure 24 qui suit :

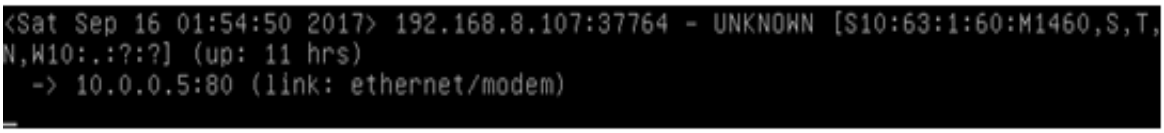

Figure 24: Résultat p0f après la connexion sur le port 80.

La connexion est bien détectée par p0f. Ce dernier offre toujours des informations précieuses sur l'attaquant tel que son adresse IP et le type de connexion utilisé pour atteindre la machine du Honeypot.

2. Fichier log de Dionaea :

La figure 25 montre une partie du fichier log de Dionaea après la connexion reçu sur le port 80.

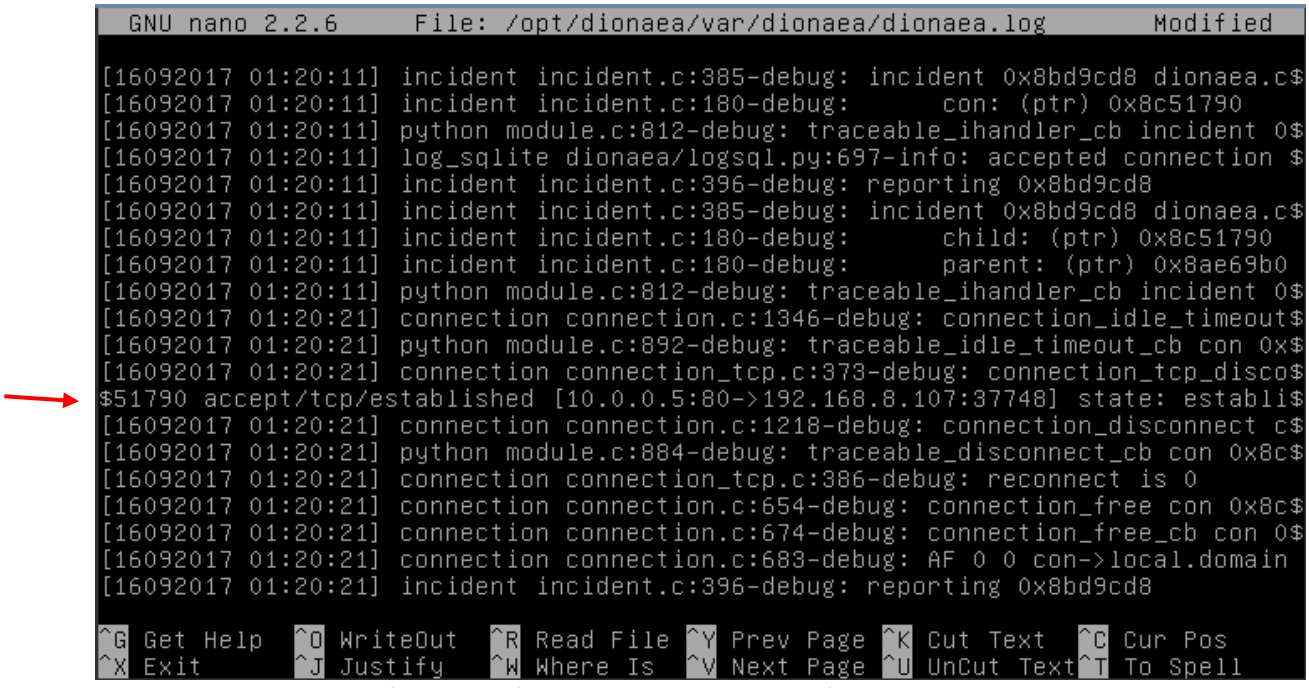

Figure 25: Résultat du fichier log de Dionaea après la connexion sur le port 80.

Sur la ligne marquée d'une flèche on voit bien qu'une connexion TCP a été reçue et acceptée. Cette connexion s'est effectuée sur le port 80 de notre machine et elle vient de la part de l'adresse IP 192.168.8.107 .

Conclusion:

Dans ce dernier chapitre nous avons commencé par donner une vue sur le réseau que nous avons mis en place en montrant bien l'emplacement du Honeynet dans le Cloud ainsi que l'emplacement du Honeywall dans le Honeynet. Nous avons ensuite enchainé avec une série de tests sur notre environnement.

Ces tests consistaient en la simulation d'attaques sur notre réseau. Chacune des attaques effectuées a été détectée par notre Honeypot Dionaea et par p0f.

En vue des résultats obtenus, nous pouvons considérer que notre objectif de départ est atteint malgré le grand manque de ressources.

Conclusion Générale:

L'essor du Cloud Computing ces dernières années a entraîné le développement d'une multitude de services associés. Ces services concernent l'utilisation de ressources informatiques à distance: applications, plateformes de développement et d'exécution et enfin infrastructures. Le modèle IaaS permet d'offrir aux clients du cloud des infrastructures virtuelles, généralement hébergées chez les fournisseurs de services. La dynamicité des environnements cloud peut impliquer des conséquences négatives sur la sécurité réseau du cloud. Pour se prémunir des menaces, ces infrastructures virtuelles sont protégées par des mécanismes de sécurité réseau comme les pare-feu virtuels et les systèmes de détection d'intrusion réseau. Les pare-feu ont pour rôle de gérer le contrôle d'accès réseau, tandis que les systèmes de détection d'intrusions doivent détecter et enregistrer les attaques survenant sur le réseau pour qu'elles soient analysées.

Dance ce contexte, nous avons proposé une approche basée sur trois phases pour renforcer la sécurité des IaaS. La première phase consiste en la création d'un environnement de Cloud Computing afin de mener notre étude sur ce dernier. Cette étape nous a donné beaucoup de fil à retordre à cause du manque de ressources ce qui nous a d'ailleurs empêchés de pousser notre étude plus loin. La seconde phase, qui est le cœur de notre travail, concerne la mise en place d'un Honeynet au sein du Cloud créée. Ceci a pour but de détecter chaque attaque sur notre machine Honeypot et d'en déduire la source et le but. Faire cela aide à comprendre les attaques prisées par les pirates et à la création de meilleures systèmes de sécurité pour les machines de production. Enfin, la dernière phase de notre approche comprend l'exécution d'attaques tests sur notre réseau afin de valider le système de sécurité mis en place. Les résultats obtenus sont encourageants, car malgré l'architecture restreinte que nous avons mis en place, le Honeynet a eu le comportement attendu et voulu.

Evidement à ce stade plusieurs perspectives nous viennent à l'esprit parmi lesquels :

- Tester cette solution sur un vrai environnement Cloud afin d'avoir une architecture complète.
- Mettre en places différents Honeypots afin d'étudier leurs différences et leurs limites pour que nous puissions, par la suite, décider de l'environnement adapté à chacun.
- L'extension de cette étude à d'autres solutions IaaS et d'autres systèmes de détection d'intrusions.
- Simuler des attaques plus poussées, voir même subir de vraies attaques pour améliorer et repousser les limites des Honeypots et donc des Honeynets.

Annexe : Création d'une instance OpenStack :

1. Avant de pouvoir installer un système d'exploitation il faut avoir l'image iso du système voulu, tout comme avant de créer une instance il faut d'abord avoir une image de machine virtuelle. Mais une image c'est quoi ? Une image de machine virtuelle est un fichier qui contient un disque virtuel doté d'un système d'exploitation bootable.

La façon la plus simple d'obtenir une image de machine virtuelle qui fonctionne avec OpenStack est de télécharger celle que quelqu'un d'autre a déjà créée. La plupart des images contiennent le paquet cloud-init pour supporter la paire de clés SSH et l'injection de données utilisateur. Pour obtenir par exemple une image ubuntu il suffi de la téléchargée depuis

http://cloud-images.ubuntu.com/

Les images téléchargées depuis ce lien ont toutes le même nom d'utilisateur qui est « ubuntu ».

2. Il faut maintenant uploader l'image au service image d'OpenStack afin que cette dernière devienne exploitable par les utilisateurs du coud.

Pour se faire il faut se connecter avec l'utilisateur démo sur le dashboard du Cloud comme le montre la figure 26.

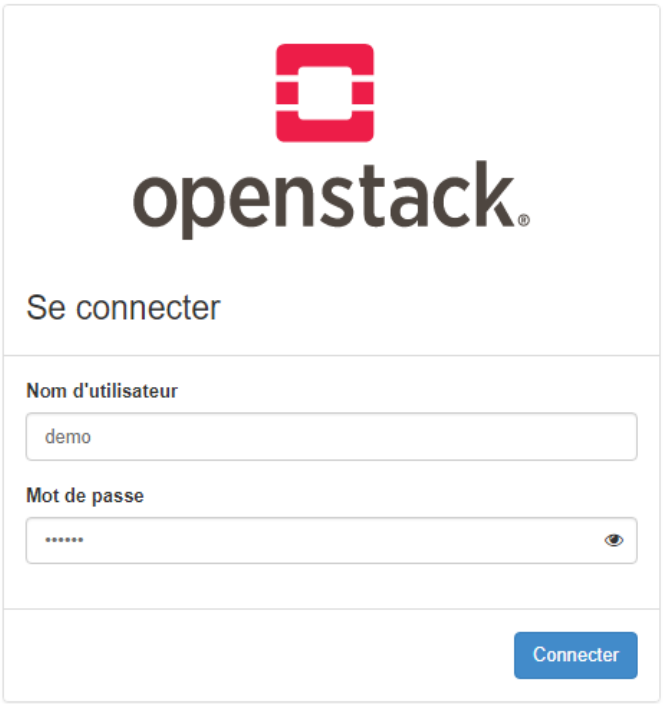

Figure 26: Connexion avec l'utilisateur demo.

Aller dans la section « images » et cliquer sur « +créer une image » montré sur la figure 27.

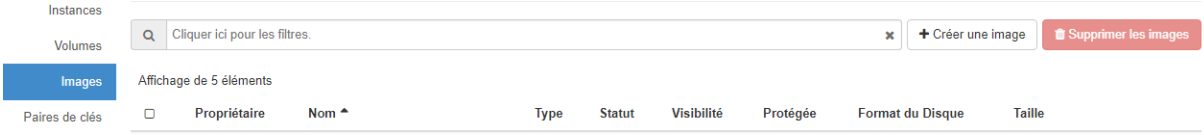

Figure 27: Créer une image.

Une boite de dialogue s'ouvrira, il suffit de remplir tous les champs et de lancer la création.

3. Maintenant que notre projet dispose d'une image nous pouvons passer à la création d'instance. Pour se faire il faut aller dans la section « instances » et cliquer sur « lancer une instance » comme le montre la figure 28.

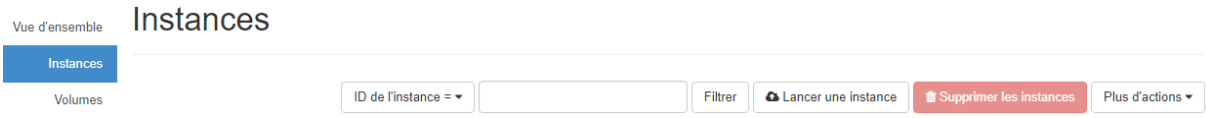

Figure 28: Créer une instance.

Quand la boite de dialogue s'ouvre il faut remplir les champs avec soins un à un. Une fois à la section « Configuration » il faudra ajouter un petit script afin de redéfinir le mot de passe de l'utilisateur ubuntu. Sans oublier de cocher la petite case pour disque de configuration comme montré dans la figure 29.

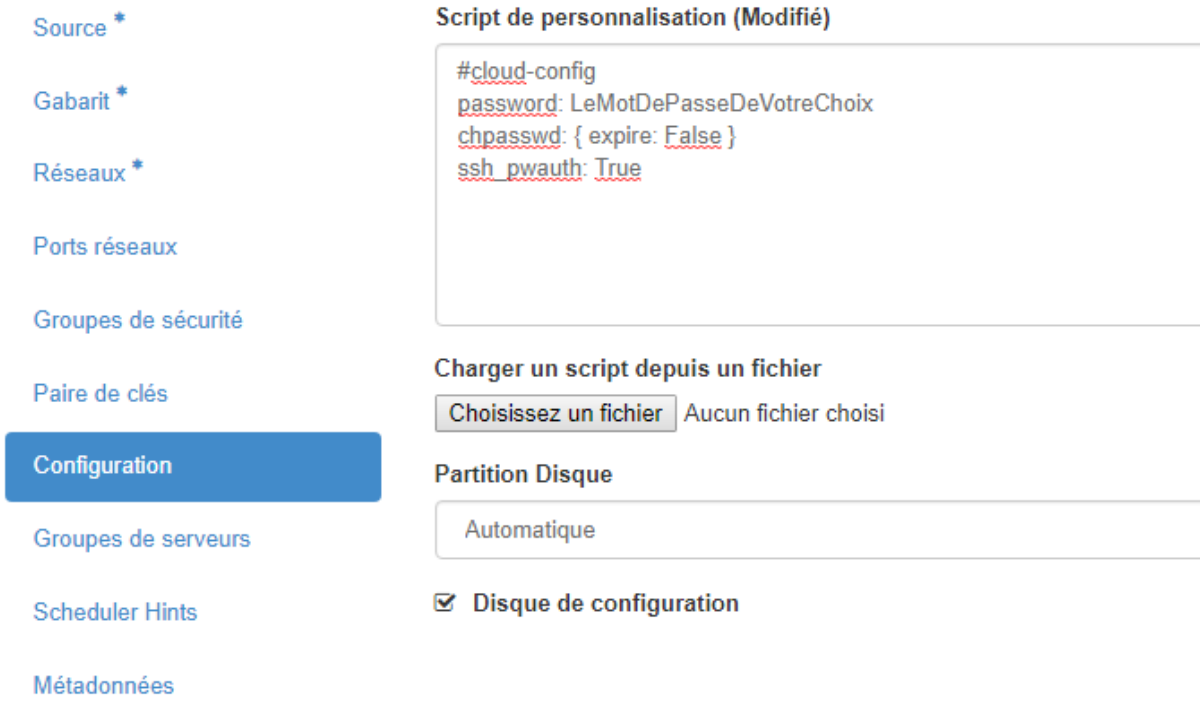

Figure 29: Ajout de script de personnalisation.

La création peut prendre quelques minutes pour passer du statut de « en construction » à « active ».

Une fois que notre instance est active nous pouvons lui associer une adresse ip flottante afin qu'elle devienne accessible à distance.

L'instance est maintenant prête à être utilisée.

Références :

[1]Wikiversité : https://fr.wikiversity.org/wiki/Cloud_computing_et_entreprise/

[2] Blog 3li : http://blog.3li.com/les-caracteristiques-essentielles-du-cloud/

[3] Lamia YOUSEFF et al. "Understanding the cloud computing landscape". In : Cloud Computing and Software Services (2010),

[4] Culture informatique : http://www.culture-informatique.net/cest-quoi-le-cloud-2/

[5] Assist : https://assist-software.net/blog/cloud-offering-comparison-between-iaas-paas-saas-baas

[6] Silicon : http://www.silicon.fr/special-cloud-3-un-monde-tout-en-aas-nouveau-87306.html

[7] Smile : http://infrastructure.smile.eu/Notre-offre/Nos-expertises/Cloud-computing-virtualisation

[8] Ivision : https://www.ivision.fr/cloud-et-virtualisation-les-differences/

[9] Khanh Toan Tran : Efficient complex service deployment in cloud infrastructure

[10] OpenStack FOUNDATION : OpenStack user stories. 2015. https://www.openstack.org/user-stories/.

[11] CloudStack : http://cloudstack.org

[12] Omar Sefraoui, Mohammed Aissaoui, Mohsine Eleuldj : Comparaison of multiple IaaS Cloud platform solutions., origine, année

[13] Daniel NURMI et al : "The eucalyptus open-source cloud-computing system". In :Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. IEEE. 2009.

[14] Borja SOTOMAYOR et al : "Virtual infrastructure management in private and hybrid clouds". In : Internet computing, IEEE 13.5 (2009).

[15] OPENNEBULA.ORG : OpenNebula architecture. 2015. http://archives.opennebula.org/_detail/documentation:rel1.2:one-architecture.png?id=documentation%3Aarchives%3Arel2.0%3Aarchitecture.

[16] TrendMicro : Cloud Security Survey - Global Executive Summary. Trend Micro Inc.report, 2011.

[17] David Stuckey, Stephen Singh, and Skip Scholl : The next generation of cloud computing. Technical report, PricewaterhouseCoopers LLP, Delaware, US, 2011.

[18] Sixto Jr Ortiz : The Problem with cloud computing standradization. Computer, 2011.

[19] Viktors Berstis : Fundamentals of Grid Computing. IBM Redbook, pages 1–28, 2002.

[20] Open Data Center Alliance Usage Model : http://www.opendatacenteralliance.org/ourwork/usagemodels

[21] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy H Katz, Andrew Konwinski, Gunho Lee, David A Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia : A view of cloud computing. Communications of the ACM, 2010.

[22] John Meegan, Gurpreet Singh, Steven Woodward, Salvatore Venticinque, Massimiliano Rak, David Harris, Gerry Murray, Beniamino Di Martino, Yves Le Roux, John McDonald, Ryan Kean, Marlon Edwards, Dave Russell, and George Malekkos :
Practical Guide to Cloud Service Level Agreements. Cloud Standard Consumer Council whitepaper.

[23] AWS Import/Export : http://aws.amazon.com/importexport/.

[24] Bureau d'expertise Technologique **Wyg**wam
Le Cloud Computing : Réelle révolution ou simple évolution ?

[25] Elie MABO, La sécurité des systèmes informatiques

[26] La sécurité des réseaux, support de cours, Mercredi, 8. novembre 2006

[27] Dominique SERET, Ahmed MEHAOUA et Neilze DORTA, « RESEUX ET TELECOMMUNICATIONS »

[28] Laurence Monaco, « Quelques définitions », 2010.

[29] Laurent Bloch et Christophe Wolfhugel, « Sécurité Informatique-principes et méthodes » (1ere édition)

[30] Laurent Poinsot «Introduction à la sécurité informatique», support de cours, Université Paris 13.

[31] Vincent Erceau & Romain Colombier, « GMSI Informatique », Projet SAS, 2011.

[32] Emonet Jeam-Burno, « Algorithme de chiffrement – mesures de performances réseaux », 2005.

[33] Rabehi Sidi Mohamed El Amine, « Mise en place d'un serveur radius sous linux pour la sécurisation d'un réseau 802.11 », Projet de fin d'étude, Université Abou Bakr Belkaid, Tlemcen-Algérie, 2011.

[34] Rachid NAIT BEKOU et Younès MOUSSAHHIL, « Etude de fiabilité et conception d'une solution VPN », Mémoire de Projet de fin d'étude, Université Mohammed V SOUSSI, Maroc, 2004.

[35] Elies Jebri, « Introduction à la sécurité», support de cours, 2008.

[36] http://www.commentcamarche.net/contents/1033-introduction-a-la-securite-informatique

[37] Laurent Bloch Christophe Wolfhugel « Sécurité informatique Principes et méthode à l'usage des DSI, RSSI et administrateurs » (2eme édition)

[38] CLÉMENT LECIGNE « les pots de miels sexy »

[39] XMCO magazine actusécu 38 , octobre 2014

[40] ENISA « Proactive Detection of Security Incidents Honeypots »

[41] Simon Clary, Adrian Winckles « How feasible is a distributed honeypot deployment in the cloud »

[42] SARNA Hélène WALLYN Laure « Dossier de sécurité : les honeypots »

[43] Iyatiti Mokube, Michele Adams « Honeypots: Concepts, Approaches, and Challenges »

[44] Gagandeep Singh, Pardeep kaur « HONEYPOTS DEPLOYMENT STRATEGIES AND LEGAL ISSUES »

[45] https://opensource.com/resources/what-is-openstack

[46] https://docs.openstack.org/ocata/install-guide-ubuntu/overview.html

[47] https://docs.openstack.org/devstack/latest/#install-linux

[48] https://www.edgis-security.org/honeypot/dionaea/

[49] http://dionaea.readthedocs.io/en/latest/index.html

[50] http://www.planet-libre.org/index.php?post_id=7813

[51] https://www.linkedin.com/pulse/dionaea-idea-malware-capturing-amirabbas-mahdavi

[52] http://lcamtuf.coredump.cx/p0f3/

[53] http://lcamtuf.coredump.cx/p0f3/README

[54]https://www.hackers-arise.com/single-post/2016/06/10/Operating-System-OS-Fingerprinting-with-p0F

[55]https://null-byte.wonderhowto.com/how-to/hack-like-pro-conduct-passive-os-fingerprinting-with-p0f-0151191/

[56] http://www.frameip.com/entete-ip/

[57] http://www.frameip.com/entete-tcp/#58-8211-fenetre

[58] Gordon “Fyodor” Lyon : The Official Nmap Project Guide to Network Discovery and Security Scanning.

[59] https://nmap.org

[60] K.C Yerrid : Netcat Starter, 2013.

[61] https://pentesterlab.com/exercises/php_include_and_post_exploitation/course

[62] Andrew Bartels, John R. Rymer, and James Staten « The Public Cloud Market Is Now In Hypergrowth »

[63] CSA « Cloud Adoption Practices and Priorities Survey Report »

[64] KPMG « (2014). Cloud Survey Report : Elevating Business in the Cloud »

[65] https://www.nist.gov/programs-projects/nist-cloud-computing-program-nccp

[66] CISCO « le cloud computing, les attributions et le rôle du département IT changent »