MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE ABDELHAMID IBN BADIS - MOSTAGANEM

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Réseaux et Systèmes**

Présenté par :

Diarra Mahamadou Daouda

THEME :

« Proposition d'une solution de sécurité contre les attaques par canal auxiliaire pour les infrastructures virtuelles dans le cloud computing »

Soutenu le : 13/06/2021

Devant le jury composé de :

Mme. N. Bahnes	MCB	Université de Mostaganem	Présidente
Mme. F. Benidriss	MCB	Université de Mostaganem	Examinatrice
Mme. F. Z. Filali	MCB	Université de Mostaganem	Encadrante

Année Universitaire 2020-2021

Dédicaces

J'ai le plaisir de dédier ce modeste travail reflétant en ma personnalité :

A mes très chers parents qui n'ont amenagé aucun effort pour que ce travail soit ainsi. J'espere qu'ils trouveront dans ce magnifique travail toutes mes gratitudes et reconnaissances.

A la mémoire de la personne que j'ai autant admiré, celle qui m'a tout donné à travers ces nobles conseils, **Nadiè Coulibaly** qu'Allah t'acceuille dans son vaste paradis.

A toute ma famille pour leur présence.

Remerciements

Au nom d'ALLAH le Tout Miséricordieux et le très Miséricordieux, et que la Paix et le Salut soit sur le bien aimé le Prophète Mohamed(PSL).

Je remercie tout d'abord Allah de m'avoir donné tout ce qu'il me fallait pour la réussite de ce travail.

Je tiens à adresser mes sincères remerciements à l'endroit de mes très chers parents ainsi qu'à toute ma famille.

Je m'en vais également à remercier mon encadrante Mme Filali Fatima Zohra pour son enthousiasme, sa patience et ses considérations ; et de m'avoir donné un tel modèle de travail si émouvant.

Je remercie Mme. F. Benidriss et Mme. N. Bahnes pour avoir accepté d'être membres de mon jury et pour leurs remarques précieuses visant à améliorer mon travail.

Mes remerciements vont à l'endroit du personnel du département informatique pour leurs encouragements ; et à tous ceux qui m'ont aidé moralement et physiquement pour la réussite de ce projet.

Table des matières

Table des figures

Liste des tableaux

Liste des symboles

ACA Attaque par Canal auxiliaire

AES Advance Encryption Standard

API Application Programming Interface

AWS Amazon Web Service

CAT Cache Allocation Technologie

CNN Convolutional Neural Networks

CPU Central processing unit

CSA Cloud Security Alliance

CSV Comma-separated values

DoS Denial of Service

*EC*2 Elastic Compute Cloud

GPU Graphical Processor Unit

IaaS Infrastructure as a Service

KVM Kernel-based Virtual Machine

L Level

LLC Last Level Cache

MITM Man-In-The-Middle

NIST National Institute of Standards and Technology

OS Operating System

PaaS Platform as a Service

PCM Performance Counter Monitoring

RAM Random Access Memory

RNN Recurrent Neural Networks

SaaS Software as a Service

SSL Secure Sockets Layer

UML Unified Modeling Language

VCPU Virtual Central processing unit

VM Machine Virtuelle

Resumé

Le cloud computing utilise de la virtualisation pour maximiser l'utilisation des ressources informatiques, par exemple de nombreuses VM s'exécutent sur une infrastructure physique partagée. Par conséquent, la co-résidence avec d'autres VM peut entraîner des risques de sécurité très élevés dans l'environnement du cloud, tels que des attaques par canaux auxiliaires. Ce type d'attaque est difficile à détecter et à prévenir, elle permet de créer des canaux auxiliaires entre VM pour obtenir des informations sensibles. Il est donc nécessaire de l'étudier en profondeur. Le but de ce travail consiste à étudier et proposer une solution contre ce type d'attaque dans les infrastructures du cloud computing.

Mots-clés : Attaque par canal auxiliaire, virtualisation, cloud computing, infrastructure.

Abstract

Cloud computing uses virtualization to maximize the use of computing resources, for example many VMs run on a shared physical infrastructure. Therefore, co-residing with other VMs can lead to very high security risks in the cloud environment, such as side channel attacks. This type of attack is difficult to detect and prevent, it creates auxiliary channels between VMs to obtain sensitive information. It is therefore necessary to study it in depth. The goal of this work is to study and propose a solution against this type of attack in cloud computing infrastructures.

Keywords : side channel attack, virtualization, cloud computing, infrastructure.

Introduction Générale

Le Cloud Computing est une technologie permettant la virtualisation des ressources telles que le processeur, les interfaces réseau, les périphériques, les disques durs et la mémoire à l'aide d'un hyperviseur.

La virtualisation matérielle mise en oeuvre dans le cloud computing, permet le partage des ressources matérielles entre plusieurs machines virtuelles de différents utilisateurs. Ce partage des ressources constitue un atout majeur des systèmes IaaS (Infrastructure-as-a-Service), dans le but d'optimiser en terme d'investissement des dépenses liées à l'achat de matériel et de logiciels et qui permet aux fournisseurs d'exploiter plus efficacement les ressources des centres de données, notamment à travers l'allocation dynamique des ressources. Cependant, le partage des ressources introduit de nouvelles contraintes de securité. L'apparition de nouvelles d'attaques propres aux infrastructures cloud computing. Ces attaques opèrent sur des ressources matérielles partagées et extraient des informations sensibles, telles que des clés cryptographiques.[47]

Le cloud computing, comme tout autre système informatique distribué, est généralement exposé à de nombreuses menaces et font l'objet d'une attaque. Les caches de processeur sont des surfaces d'attaque populaires qui mènent à des attaques de canaux auxiliaire entre VM. Pour se prémunir de cette attaque reposant sur le partage des ressources. Nous avons proposé solution de détection et d'atténuation pour contrer cette attaque, elle permet de détecter s'il y a une anomalie. Si oui, on passe à l'atténuation sinon pas de problème à signaler .

Dans le cadre de ce mémoire, nous avons structuré notre travail comme suit : une Introduction Générale, un premier chapitre sur le Cloud computing en général, un deuxième chapitre sur l'attaque par canal auxiliaire, un troisième chapitre sur l'analyse et la conception, un quatrième chapitre pour la réalisation et enfin une conclusion générale pour clôturer.

Chapitre 1

Concepts et Problématiques

I. Introduction

Dans le cadre de l'évolution de la technologie de l'information un nouveau paradigme informatique fait son apparition afin de révolutionner le monde de l'informatique en offrant des services d'accès sur demande. Le cloud computing est un paradigme émergeant dans le monde de l'informatique fournissant des services telle qu'une infrastructure à grande échelle pour le calcul de haute performance qui s'adapte dynamiquement à l'utilisateur et les besoin de l'application.

Dans ce chapitre nous aborderons des concepts liés au cloud computing et les problèmes de sécurité auxquels il est confronté.

II. Généralité sur le cloud computing

1. Définitions

Selon le National Institute of Standards and Technology (NIST), le cloud computing est un modèle permettant un accès réseau omniprésent, pratique et à la demande à un pool partagé de ressources informatiques configurables (par exemple, réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement provisionnés et libérés avec un effort de gestion minimal ou interaction avec le fournisseur de services. Ce modèle cloud est composé de cinq caractéristiques essentielles, trois modèles de service et quatre modèles de déploiement. [1]

Selon Amazon Web Service(AWS), le cloud computing est la mise à disposition de ressources informatiques à la demande via Internet, permettant d'accéder aux guise de services

technologiques, tels que la puissance de calcul, le stockage et les bases de données d'un fournisseur. [38]

A partir de ces définitions, nous pouvons donc déduire de manière simple que le Cloud Computing est une dématérialisation des ressources informatiques.

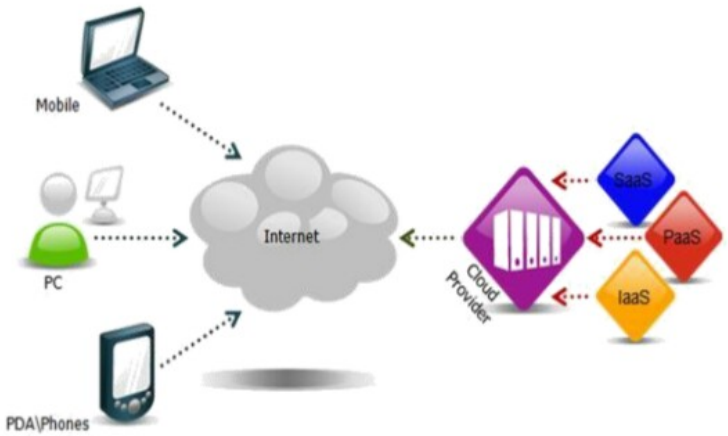

FIGURE 1.1 – Architecture du cloud computing [10]

2. Caractéristiques essentielles du cloud computing

Selon NIST, le cloud computing est un modèle composé de cinq(5) caractéristiques essentielles à savoir :[1, 27]

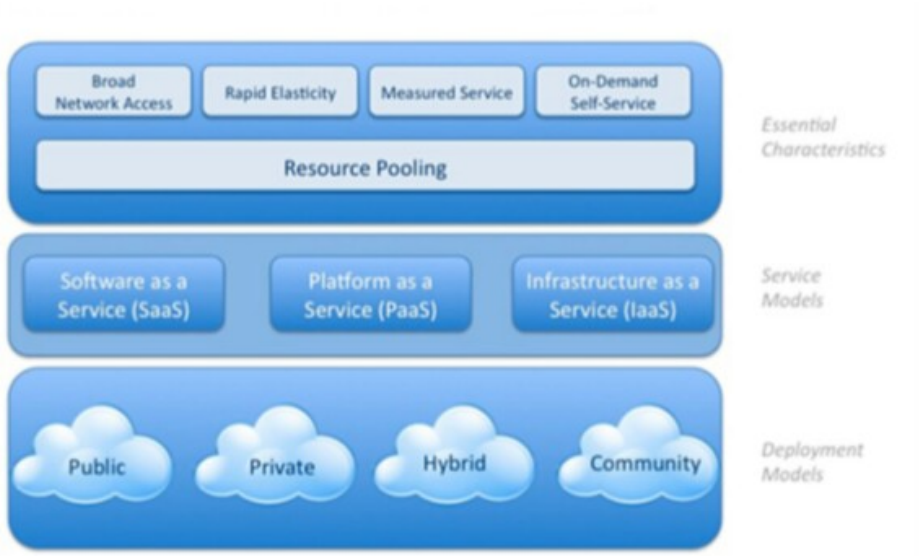

FIGURE 1.2 – Le modèle visuel du cloud computing de NIST [4]

2.1. Libre-service à la demande

L'utilisation des services et ressources est entièrement automatisée et c'est l'utilisateur, au moyen d'une console de commande, qui met en place et en gère la configuration à distance.

2.2. Elasticité rapide

Les ressources peuvent être provisionnées et libérées de manière élastique et automatique, pour répondre rapidement et de manière extensible à la demande.

2.3. Large accès réseau

Les services sont accessibles depuis le réseau Internet par des équipements traditionnels et hétérogènes, légers ou lourds.

2.4. Mise en commun de ressources(Mutualisation)

La mutualisation consiste à regrouper les ressources informatiques du fournisseur et les utiliser pour servir plusieurs clients selon un modèle de co-résidence, avec des ressources dynamiquement allouées selon les demandes.

2.5. Paiement à l'usage

L'utilisation du service cloud est facturé à l'usage, en fonction de la consommation du service.

3. Les Modèles de déploiements du cloud computing

Les services cloud peuvent être déployés de quatre(4) manières suivantes : cloud privé, cloud public, cloud communautaire, cloud hybride .[1, 27, 39]

3.1. Cloud privé

L'infrastructure cloud est mise à disposition pour une utilisation exclusive par une seule organisation comprenant plusieurs consommateurs (par exemple, des unités commerciales).

3.2. Cloud public

L'infrastructure cloud est configurée pour une utilisation ouverte par le grand public. Il peut être détenu, géré et exploité par une entreprise, un universitaire ou une organisation gouvernementale, ou une combinaison des deux. Il existe dans les locaux du fournisseur de cloud.

3.3. Cloud communautaire

L'infrastructure cloud est mise à disposition pour une utilisation exclusive par une communauté spécifique de consommateurs d'organisations partageant des préoccupations (par exemple, la mission, les exigences de sécurité, les politiques et les considérations de conformité).

3.4. Cloud hybride

L'infrastructure cloud est une composition d'au moins deux infrastructures cloud distinctes (privées, communautaires ou publiques) qui restent des entités uniques, mais sont liées entre elles par une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications (par exemple, l'éclatement du cloud pour l'équilibrage de charge entre des nuages).

4. Les modèles de services dans le cloud computing

Une infrastructure cloud est un ensemble de matériels et de logiciels. Cette infrastructure peut être considérée comme contenant à la fois une couche physique et une couche d'abstraction. La couche physique comprend les ressources matérielles nécessaires pour prendre en charge les services cloud fournis et comprend généralement des composants de serveur, de stockage et de réseau. La couche d'abstraction se compose du logiciel déployé sur la couche physique, qui manifeste les caractéristiques essentielles du cloud. Conceptuellement, la couche d'abstraction se trouve au-dessus de la couche physique.[1]

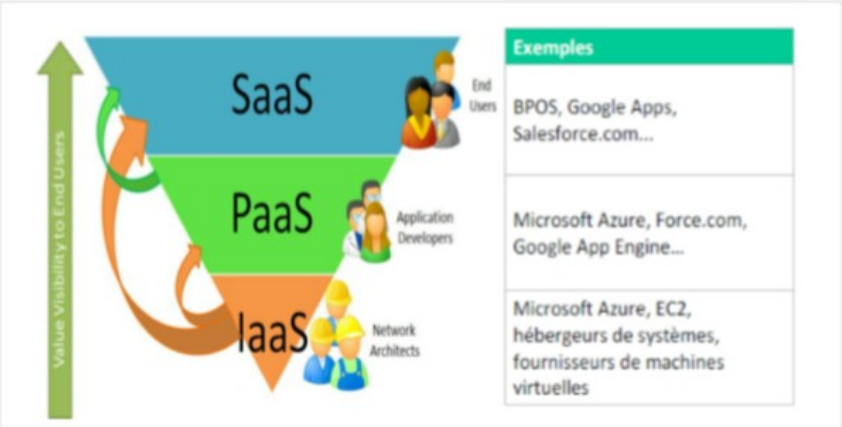

FIGURE 1.3 – Modèles de services de cloud computing [31]

4.1. Saas(Software as a Service)

La capacité fournie au consommateur consiste à utiliser les applications du fournisseur s'exécutant sur une infrastructure cloud. Les applications sont accessibles à partir de divers appareils clients via une interface de client léger, comme un navigateur Web (par exemple, une messagerie Web), ou une interface de programme.

Exemple : Un logiciel tel qu'un logiciel de comptabilité.

4.2. Paas(Platform as a Service)

La capacité fournie au consommateur consiste à déployer sur l'infrastructure cloud des applications créées ou acquises par le consommateur créées à l'aide de langages de programmation, de bibliothèques, de services et d'outils pris en charge par le fournisseur.

Exemple : Une plate-forme tel qu'un environnement de développement.

4.3. Iaas(Infrastructure as a Service)

La capacité fournie au consommateur est de fournir le traitement, le stockage, les réseaux et d'autres ressources informatiques fondamentales où le consommateur est capable de déployer et d'exécuter des logiciels arbitraires, qui peuvent inclure des systèmes d'exploitation et des applications.

Exemple : L'infrastructure telle que les serveurs virtuels .

	Saas	Paas	Iaas
Description	Une application complète en tant que service à la demande.	Une plateforme de développement et de déploiement d'applications	Matériel et logiciels associés
Avantages	Réduction des dépenses d'exploitation	Réduction des coûts et de la complexité de l'achat et de la gestion des outils.	Haute flexibilité, faible coût, accès aux dernières technologies
Exemple	Facebook	GoogleApp engine	Amazon EC2

TABLE 1.1 – Récapitulatif du Modèles de services cloud [8]

5. les avantages et inconvénients du cloud computing

Le Cloud Computing offre beaucoup d'avantages et d'inconvénient.[39]

5.1. Avantages

1. **L'intérêt économique :** Permet d'optimiser le coût d'investissement dans l'achat de logiciels ou plate-formes coûteuses.
2. **La facilité d'administration :** L'entreprise n'a plus besoin de se soucier de l'hébergement du service, de sa maintenance logicielle, des mises à jour, de la sécurité du service, etc.
3. **La souplesse :** Grâce à la mutualisation du service entre nombreux clients, le fournisseur du service cloud autorise une très grande souplesse. Ce qui rend simple et sans conséquence les évènements tels que la nécessité de monter/baisser en charge très rapidement, ou tout simplement ne plus utiliser le service.
4. **Le gain de productivité :** les entreprises peuvent davantage se concentrer sur l'utilisation du service plutôt que son administration.

5.2. Inconvénients

1. **La perte de contrôle :** Comme l'hébergeur ou éditeur du service cloud se charge de tout, nous n'avons plus aucun contrôle sur le fonctionnement du service, ce qui entraine une manque de confiance à l'hébergeur.
2. **Le coût :** le cloud permet généralement de réaliser des économies, cela ne reste vrai que si l'entreprise se dote de la volonté d'une réelle maîtrise des coûts. La facilité d'ajouter de nouvelles ressources peut conduire à un véritable gaspillage.

3. **Les risques de migration :** De nombreux services cloud ont la particularité, volontairement ou non, d'utiliser des technologies ou environnements non standards, rendant ainsi complexe voire impossible la migration vers un hébergeur ou éditeur concurrent.
4. **Sécurité :** Les risques d'attaque et de perte de confidentialité sont augmentés dans le cloud car les données du client sont hébergées en dehors de sa porté. Cela peut donc poser un risque potentiel pour le client de voir ses données mal utilisées ou volées.

III. Virtualisation

1. Définition

La virtualisation consiste à faire fonctionner des ressources informatiques virtuelles à partir d'une machine physique réelle.[3]

2. Les différents types de virtualisations

Il y a différent types de vitualisations : virtualisation complète, para-virtualisation et virtualisation assistée par matériel.[9]

1. **Virtualisation complete**

   La virtualisation complète consiste à émuler un environnement matériel complet sur les machines virtuelles c'est-à-dire toutes les ressources système sont virtualisées telles que les processeurs, la mémoire et les périphériques d'E / S, et exécutent un système d'exploitation non modifié.

2. **Paravirtualisation**

   La para-virtualisation est une technique qui n'est pas totalement transparente du point de vue d'une VM permettant l'utilisation d'un système d'exploitation hôte léger pour la virtualisation.

3. **Vrtualisation assistée par matériel**

   La virtualisation au niveau matériel est prise en charge par Intel et AMD directement au niveau du processeur, par les techniques VT-X et AMD-V.

3. Les differents types d'hyperviseurs

Un hyperviseur est un gestionnaire de machine virtuelle.En outre, il existe principalement deux types d'hyperviseurs : hyperviseur de type 1(bare metal) et hyperviseur de type 2(hosted).[27]

1. **Hyperviseur de type 1 :** L'hyperviseur s'exécute directement sur les ressources matérielles et implémente certaine fonctionnalités des noyaux des systèmes d'exploitations.
   Exemple : Xen, Vmware ESXi, Microsoft Hyper-V.

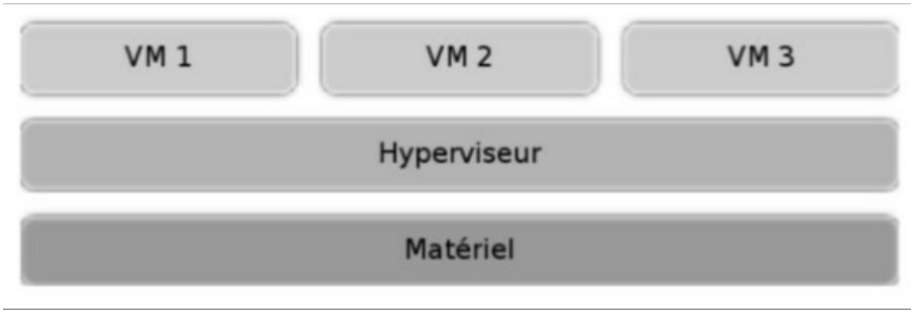

FIGURE 1.4 – Hyperviseur de type 1 [27]

2. **Hyperviseur de type 2 :** L'hyperviseur est réduit à un logiciel pour fonctionner sur un système d'exploitation appelé système hôte.
   Exemple : Oracle VirtualBox, Vmware Workstation.

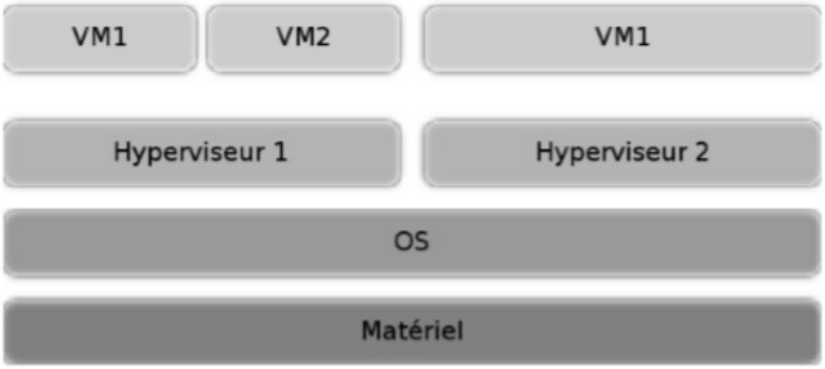

FIGURE 1.5 – Hyperviseur de type 2 [27]

IV. Problématiques de sécurité

Le cloud computing, comme tout système informatique distribué, est exposé à de nombreuses menaces. Cependant, un certain nombre de questions se posent sur la sécurité de l'in-

formation des différentes menaces que fait face le Cloud computing, et surtout les données stockées afin de garantir la confiance au service cloud.

Ces questions peuvent être : Où sont les données physiquement ? Sous quelle juridiction ? Les données sont sur le même serveur que les données de mes concurrents ? Comment puis-je être sûr que mes concurrents ne puissent pas accéder à mes données ? Que deviennent mes données lorsque je me désabonne d'un service Cloud ? Comment puis-je m'assurer que mes données sont supprimées ? Quel niveau de service puis-je avoir ? Si ce niveau n'est pas atteint, contre qui et comment puis-je réclamer ?

Ainsi, la sécurité du cloud est aujourd'hui'hui une préoccupation très importante des fournisseurs et utilisateurs. Pour se prémunir des attaques reposant sur l'utilisation du cloud computing, des mécanismes de sécurité sont déployés pour protéger les données hébergées dans les infrastructures virtuelles que nous verrons plus tard .

1. Les principales ménaces dans un environnement cloud computing

La Cloud Security Alliance (CSA) a défini sept grandes classes de menaces pour le cloud :[7]

1. Abus et usage néfaste du cloud.
2. APIs et interfaces non sécurisées.
3. Malveillance interne.
4. Problèmes liés aux technologies de partage.
5. Perte ou fuite de données.
6. Détournement de compte ou de service.
7. Profil de risque inconnu.

2. Les vulnérabilités dans un environnement cloud computing

Les services cloud présentent des vulnérabilités dans un environnement cloud. Nous allons en citer quelques unes :[2]

1. **Les utilisateurs malveillants :** Cloud Computing a besoin des profils utilisateurs de haut niveau pour son administration. Un administrateur système aura des privilèges complets sur différentes ressources de différents clients. Un utilisateur malveillant qui compromet la sécurité du système avec succès et saisie une session administrateur obtiendra l'accès à de nombreuses informations clientes.

2. **La suppression dangereuse ou incomplète des données :** La réutilisation des ressources matérielles est très commune dans le Cloud Computing. Un nouveau client peut, par exemple, être affecté d'une section de stockage dans lequel les données d'un autre client y étaient. Cela peut entraîner un risque de perte de confidentialité, si les données précédentes n'ont pas été supprimées complètement et en toute sécurité.
3. **La protection des données :** Pour le client de services Cloud Computing, la protection des données est difficile. Il est très difficile de sécuriser les données qui se trouvent réparties dans plusieurs emplacements. S'assurer que les données sont traitées correctement est également compliqué pour le client parce que le contrôle sur les transferts de données est hors de sa porté.

3. Les attaques dans un environnement cloud computing

Il y a plusieurs techniques d'attaques dans un environnement cloud, nous énumérons quelques une d'entre elles : [22, 8]

1. **Attaque DoS (Denial of Service) :** C'est une attaque qui consiste inonder le Cloud par un grand nombre de requêtes via les machines zombies à travers l'internet afin de rendre le service indisponible.[6]
2. **Attaques par force brute :** Tentatives d'exploit comptant un grand nombre de combinaisons pour trouver une faille.
3. **Attaque Porte dérobée (Backdoor) :** C'est une attaque qui permet d'accéder à distance au système par l'utilisation de portes dérobées. L'adversaire peut être en mesure de contrôler les ressources de la victime et peut en faire un zombie pour tenter une attaque DDoS.
4. **Attaque par injection de malware :** C'est une attaque considérable qui tente d'injecter un programme malveillant ou de mettre en œuvre une machine virtuelle malveillante dans le Cloud.
5. **Attaque de l'homme du milieu MITM (Man-In-The-Middle) :** C'est une attaque permettant de s'introduire dans une communication en cours entre deux utilisateurs du Cloud (ou de n'importe quel système), dans le but d'acquérir des informations sur des données transférées entre eux.

4. Problèmes de sécurité de l'hyperviseur

La virtualisation est l'un des éléments clés du cloud computing. Les entreprises peuvent utiliser la virtualisation pour réduire les dépenses en capital sur le matériel serveur et augmenter l'efficacité opérationnelle.

Cependant, la virtualisation apporte tous les problèmes de sécurité du système d'exploitation fonctionnant en tant qu'invité, ainsi que de nouveaux problèmes de sécurité concernant la couche hyperviseur et de nouvelles menaces spécifiques à la virtualisation, des attaques entre VM(machine virtuelle), des problèmes de performances liés au processeur et à la mémoire utilisés pour la sécurité.

Il existe principalement deux types d'attaque sur la virtualisation : évasion de la machine virtuelle et Rootkit dans l'hyperviseur.[32]

1. **Evasion de la machine virtuelle :** cette attaque consiste à casser la couche d'isolation afin d'exécuter les privilèges root de l'hyperviseur à la place des privilèges machine virtuelle. Cela permet à un attaquant d'interagir directement avec l'hyperviseur. Par conséquent, l'évasion de la machine virtuelle à l'isolement est assurée par la couche virtuelle par le biais de cette attaque, un attaquant obtient l'accès au système d'exploitation de l'hôte et aux autres machines virtuelles en cours d'exécution sur la machine physique.
2. **Rootkit dans l'hyperviseur :** Cette attaque consiste à créer un canal secret pour exécuter du code non autorisé dans le système afin d'acquérir le contrôle sur tout le fonctionnement de l'hôte, des machines virtuelles en cours d'exécution et des activités présentent dans le système.

V. Conclusion

En effet, le Cloud computing offre différents services aux utilisateurs ainsi qu'aux entreprises. Cependant, la confidentialité des données personnelles et la confiance restent les principales préoccupations de tous les fournisseurs de services cloud. Ce qui peuvent empêcher les utilisateurs d'adopter massivement une solution cloud computing à travers des méthodes employées, qui sont vulnérables à de nombreuses attaques, ce qui engendre l'insécurité des données des utilisateurs.

Chapitre 2

Attaque par canal auxiliaire

I. Introduction

La confidentialité et l'intégrité sont les éléments clés de la sécurité de l'information. L'infrastructure en tant que service (IaaS) est un service de cloud computing où le fournisseur de cloud loue des ressources informatiques aux clients du cloud en termes de VM. Ces VM peuvent s'exécuter sur la même machine physique et partager toutes ses ressources matérielles sous-jacentes. Ce partage peut permettre à une machine virtuelle malveillante d'extraire des informations sensibles à partir de machines virtuelles co-résidentes, ce qui engendre un problème sérieux de sécurité freinant l'adoption du service cloud.

Dans ce chapitre, nous présenterons les attaques par canal auxiliaire et les principes permettant d'extraire les informations d'une machine virtuelle vers une autre.

II. Définition

Selon Guillaume Duc, une attaque par canal auxiliaire peut être défini comme suit : "Une application malicieuse s'exécutant sur une première machine virtuelle pouvant voler, en ciblant les variations de temps de calcul introduit par les caches des processeurs, des données manipulées par une seconde machine virtuelle tournant sur une même machine physique".[12]

Alors, nous pouvons déduire que l'attaque par canal auxiliaire est une attaque qui consiste pour une machine virtuelle extraire des information d'une autre machine virtuelle co-résidente.

III. Le principe de fonctionnement

1. Fonctionnement

Les attaques par canaux auxiliaires sont identifiées comme une attaque très sophistiquée dans le cloud computing. Dans la mesure où il existe des ressources matérielles partagées, l'attaque par canal auxiliaire exploite les informations obtenues à partir, par exemple, de l'unité centrale (CPU) et de la mémoire cache.

Dans ce type d'attaque entre VM, un adversaire place une VM malveillante co-résidente avec la VM cible afin qu'elle partage les mêmes ressources matérielles. Ensuite, l'attaquant extrait des informations utiles telles que les clés cryptographiques de la VM cible et les utilise pour l'écoute du trafic.

Grâce à l'attaque par canal auxiliaire, un attaquant partageant le même cache que la victime peut surveiller le comportement d'accès au cache de la victime. Par exemple, l'attaquant est capable de surveiller les informations de synchronisation du cache en mesurant l'exécution de différentes opérations sur la machine virtuelle de la victime. En règle générale, l'attaquant exploite les synchronisations de la mémoire cache de haut niveau partagée.[18]

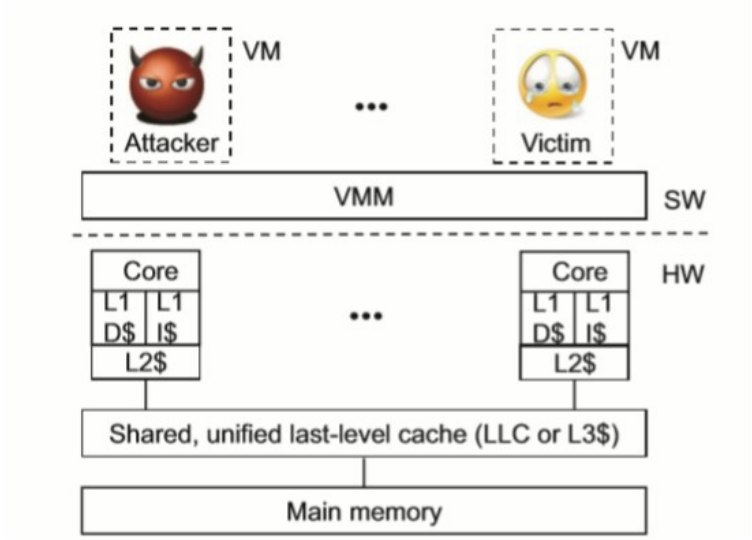

FIGURE 2.1 – Fonctionnement de l'ACA [35]

1.1. Types de canaux auxiliaires

Un canal auxiliaire peut être établi soit entre les processus ou entre les VM.[9]

1. **Entre processus :** Ce type de canal est généralement établi entre deux processus, c'est-à-dire un processus espion et un processus victime, qui s'exécutent sur le même système d'exploitation, généralement sur le même cœur. Le processus d'espionnage doit être exécuté en parallèle avec le processus victime et les deux doivent être bien synchronisés.

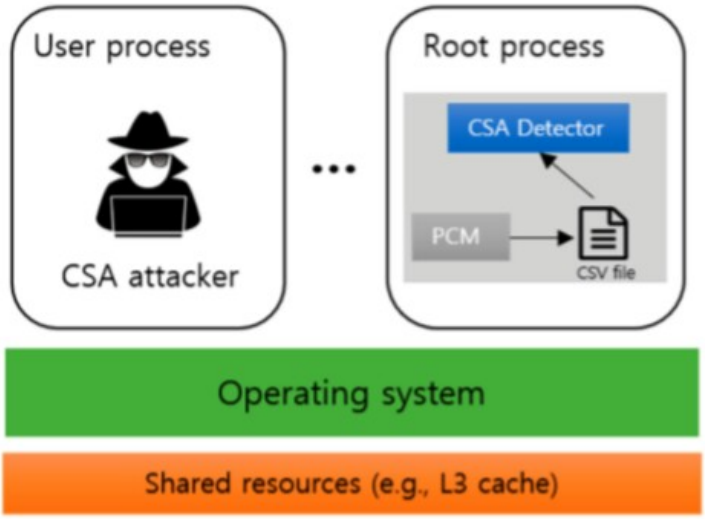

FIGURE 2.2 – Attaque par canal auxiliaire dans un environnement OS [19]

2. **Entre VM :** Ce canal est créé entre deux VM s'exécutant sur le même cœur de processeur (co-résidence). De plus, ce type de canal doit supporter plus de bruit que les canaux inter-processus. Enfin, un canal auxiliaire peut être créé via un réseau, produisant un canal plus bruyant en raison des propriétés du réseau, telles que la latence.

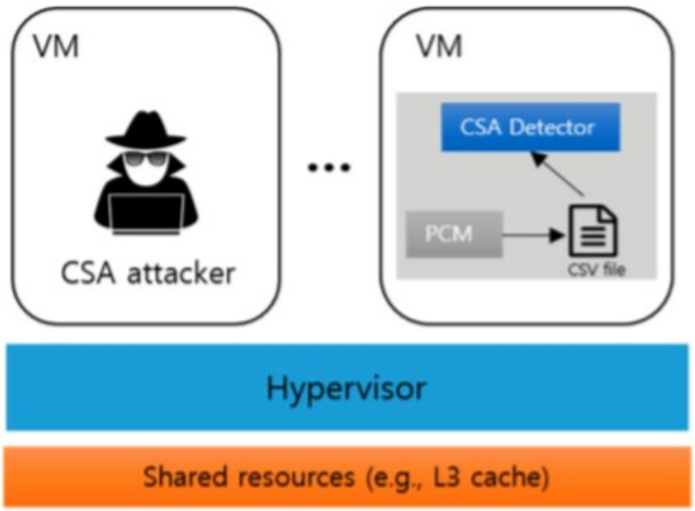

FIGURE 2.3 – Attaque par canal auxiliaire dans un environnement virtualisé [19]

1.2. Catégories d'attaque par canal auxiliaire

Elle est divisé en deux grandes catégories :

1. **Attaques passives :** Un attaquant observe les activités de la cible sans effectuer de modifications sur la cible afin d'obtenir des informations.
2. **Attaques actives :** Les attaques actives modifient l'environnement de la cible d'attaque en forçant la cible à effectuer des opérations anormales.

IV. Techniques d'attaque par canal auxiliaire

Il existe différentes techniques d'attaque en fonction du type de canal exploité sont : attaques par timing, cache, électromagnétiques et surveillance de l'alimentation. Les attaques électromagnétiques et les attaques de surveillance de l'alimentation sont plus applicables aux appareils physiques tels que les cartes à puce. Les attaques basées sur le cache et le timing sont les principales attaques logicielles applicables dans le cloud computing en raison du partage des ressources et des techniques de virtualisation. [9]

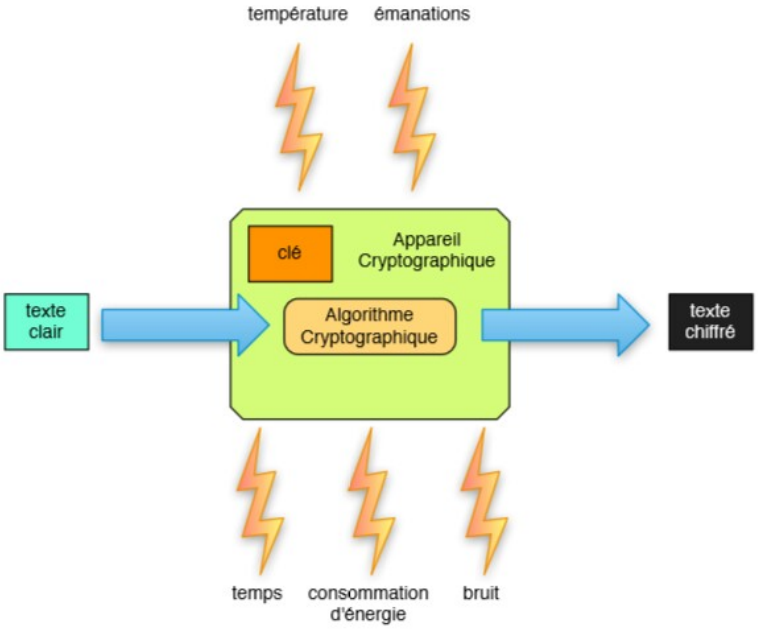

FIGURE 2.4 – Les techniques d'attaque par canal auxiliaire [28]

1. Attaque par canal auxiliaire basée sur la cache

Le cache du processeur est la principale source d'informations sur laquelle les attaquants profitent pour effectuer des attaques par canal auxiliaire.

1.1. Architecture de cache

D'après les chercheurs le processeur est la ressource la plus ciblée par les attaquants. Le microprocesseur moderne a plusieurs niveaux de cache : niveau 1, niveau 2 et niveau 3. Dans un processeur avec plusieurs cœurs et trois niveaux de cache, chaque cœur de processeur a son propre cache L1 et L2 séparé tandis que le cache L3 est partagé entre tous les cœurs du processeur.

Par conséquent, l'architecture partagée du cache L3 entre différents cœurs crée une opportunité pour un utilisateur malveillant de l'exploiter et de lancer une attaque de canal auxiliaire. La fonction de base des caches CPU est de mettre en mémoire tampon les données demandées à la mémoire principale, en raison des performances de compromis entre les mémoires rapides à lentes et petites à grandes, alimentent la CPU lorsqu'elle fonctionne sur les données demandées.

Dans un cas concret, lorsque le processeur a besoin de données, il vérifie respectivement

les caches L1, L2 et L3. Si les données demandées sont présentes à n'importe quel niveau, les données sont lues et un accès au cache se produit. Cependant, si les données ne sont disponibles à aucun niveau du cache, un échec de cache se produit et le processeur obtient les données de la RAM et les place dans le cache pour le prochain appel. [33, 36]

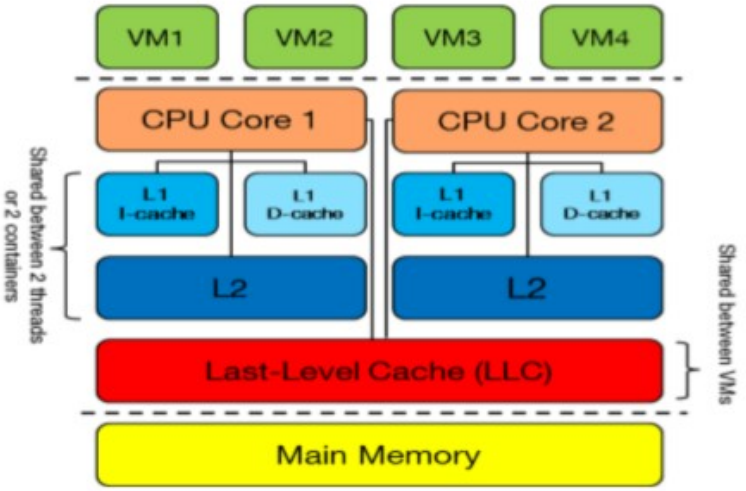

FIGURE 2.5 – L'architecture du cache [9]

1.2. Organisation du cache

Le cache peut être organisé en trois catégorie : Full associative, Direct Mapping et set associative.[33]

1. **full-Associative** Dans ce cas, un bloc de mémoire peut occuper n'importe laquelle des lignes de cache dans le cache c'est-à-dire n'importe quelle ligne de la mémoire principale peut être située à n'importe quel emplacement du cache.

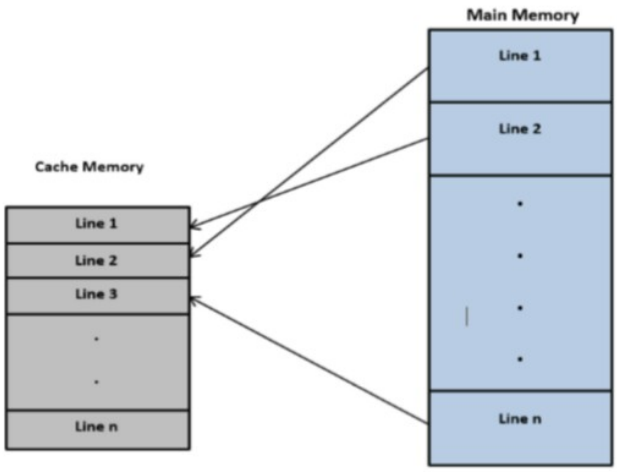

FIGURE 2.6 – Full Associative

2. **Direct-Mapping** Le mappage direct dans le cache est également appelé cache associatif unidirectionnel. Dans ce cas, chaque bloc de mémoire a un emplacement fixe dans le cache, c'est-à-dire qu'ils ne peuvent occuper qu'une seule ligne de cache spécifique.

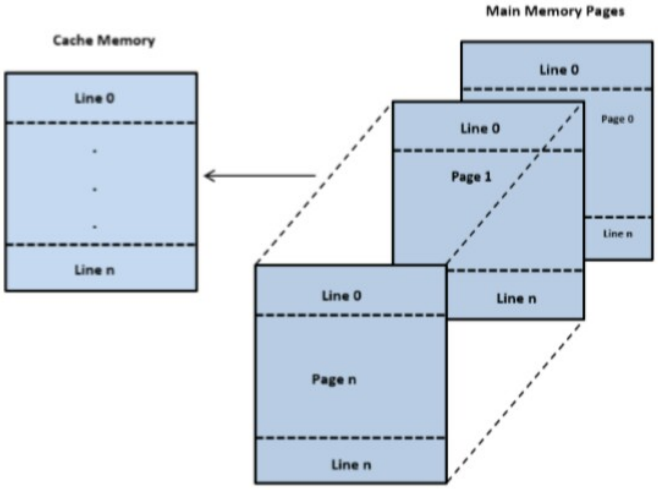

FIGURE 2.7 – Direct Mapping

3. **Set-Associative** C'est une conception qui divise le cache en partitions de taille égale appelées ensembles de cache, chacune allouant n lignes de cache. Dans ce cas, un bloc mémoire est contraint d'occuper l'une des n lignes de cache d'un ensemble fixe.

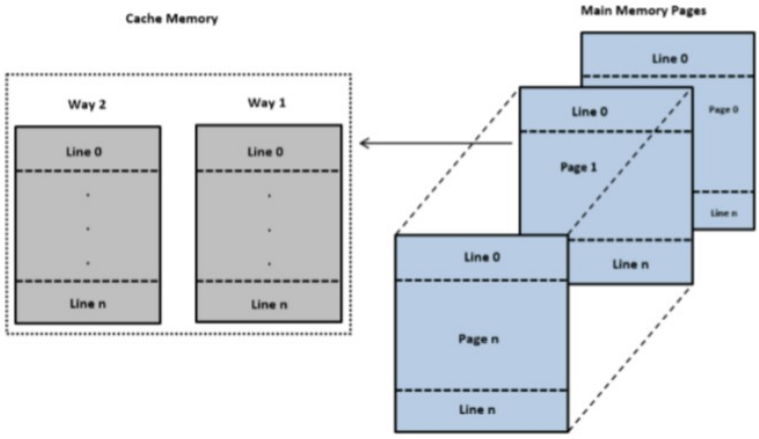

FIGURE 2.8 – set Associative

1.3. Les Types d'attaques par canal auxiliaire côté cache

L'attaque par canal auxiliaire est divisée en trois classes : pilotée par le temps, pilotée par trace et pilotée par l'accès .[11, 37]

1. **Attaque pilotée par l'accès(Access-driven attack()**

   Dans ce type d'attaque, l'attaquant tente de trouver une relation entre un processus de chiffrement et l'accès aux lignes de cache. De plus, il exécute un programme d'espionnage sur la machine physique qui l'héberge et la victime, afin d'obtenir des informations sur l'ensemble de cache auxquels la victime a accédé. Les informations sont extraites de l'ensemble du cache accédés lors de l'exécution du processus cible.

2. **Attaque pilotée par la trace (Trace-driven attack)**

   C'est une attaque cherchant à obtenir des informations relatives au nombre total d'échecs ou de succès de cache pour un processus ou une machine ciblé. Un attaquant peut capturer le profil des activités du cache pendant un cycle de chiffrement en termes d'accès ratés et réussis de la victime. Après avoir capturé suffisamment d'échantillons, une phase hors ligne permet de déduire les données secrètes qui sont utilisées par le processus victime. Les informations sont obtenues en surveillant directement le cache, en accédant au profil du cache lors de l'exécution du processus cible.

3. **Attaque pilotée par le temps (Time-driven attack)**

   Dans ce type d'attaque, un attaquant vise à mesurer l'ensemble des temps d'exécution des opérations cryptographiques avec une clé fixe. L'ensemble des temps d'exécution est influencé par la valeur de la clé. Ainsi, l'attaquant exécutera un cycle de cryptage ; expul-

ser un ensemble de cache sélectionné en y écrivant ses propres données et en mesurant le temps nécessaire à un cycle de cryptage par la victime. Le temps nécessaire au chiffrement dépend des valeurs du cache au démarrage du chiffrement. Les informations sont apprises en observant le profil de synchronisation pour plusieurs exécutions d'un chiffrement cible.

1.4. Techniques d'attaque des canaux auxiliaires basée sur le cache

L'attaque par canal auxiliaire basée sur le cache a plusieurs techniques qui sont : Prime et Probe, Flush et Reload, Flush et Flush, Evict et Time.

1. **Prime et Probe**

   Dans cette technique, un processus attaquant surveille le processus victime en remplissant le cache du processeur avec ses données afin de vérifier quelles lignes de cache de l'attaquant ont été expulsées par les données de la victime. le processus attaquant utilise une boucle occupée qui dort pendant un temps spécifique à chaque itération pour réveiller le processus attaquant et mesurer la variation du temps d'accès à ses lignes de cache. Un temps plus long indique que la ligne de cache est expulsée par la victime et doit être chargée à partir d'un niveau de mémoire supérieur. Cette technique peut être appliquée à tous les niveaux de mémoire. [36, 37]

2. **Flush et Reload**

   Cette technique est l'inverse de prime et probe. L'attaquant et la victime doivent avoir accès aux mêmes données simultanément ; c'est une fonctionnalité de la bibliothèque partagée. L'attaquant cible une gamme d'adresses dans une bibliothèque partagée, qui est chargée dans la mémoire principale, et vide une ou plusieurs lignes du cache référencées dans la plage d'adresses. L'attaquant doit être sûr que les lignes du cache ont été supprimées de la hiérarchie de cache (L1, L2 et L3). Après le vidage, l'attaquant attend que la victime accède à certaines données de la plage d'adresses.Lorsque la victime tente d'accéder à la ou aux lignes de cache vidées, il doit y accéder en mémoire principale, en raison du vidage effectué par l'attaquant. Enfin, l'attaquant scanne et encode les accès horaires ; un temps plus court indique que la ou les lignes de cache ont été récemment accédées par la victime. [36, 37]

3. **Flush et Flush**

   Dans cette technique, l'attaquant s'appuie sur la variation temporelle du vidage des séries

dans l'instruction plutôt que sur la surveillance des accès à la ligne de cache.[36]

4. **Evict et Time**

   Dans cette technique, on suppose qu'une bibliothèque partagée est liée à la fois à l'attaquant et au programme victime en même temps. L'attaquant surveille les lignes de cache synchronisées avec la baie. L'attaquant trouve d'abord le temps moyen nécessaire pour un cryptage, puis déclenche la fonction de cryptage et expulse les lignes de cache, qui ont déjà touché la baie. Après l'expulsion, l'attaquant déclenche une série de cryptages et les mesure. Si un appel de chiffrement prend plus de temps que la durée moyenne, cela indique que la ou les lignes de cache expulsées ont été accédées récemment par la victime.[36, 37]

2. Timing attacks

Dans ce type d'attaque, un attaquant peut tenter, par exemple, d'analyser le temps nécessaire pour exécuter des algorithmes cryptographiques afin d'extraire des clés de chiffrement. Ces attaques sont mieux applicables à des dispositifs tels que les cartes à puce, elles peuvent également cibler des crypto-systèmes basés sur le réseau et des implémentations logicielles d'algorithmes cryptographiques, tel que OpenSSL. Les attaques par timings peuvent être locales ou distantes :

- Dans le cadre d'une attaque locale, un programme espion est exécuté sur la même machine qu'un programme victime.
- Lors d'une attaque à distance, la victime et l'attaquant sont hébergés sur des machines différentes, dans un réseau local ou dans le cloud. Les attaques à distance sont notoirement difficiles à combattre en raison du bruit induit par le réseau. Ce type d'attaque est également applicable aux crypto-systèmes basés sur le cloud qui sont répartis entre plusieurs machines virtuelles.Les attaques de synchronisation basées sur le cache exploitent la mémoire cache comme moyen d'attaquer localement un crypto-système. Le temps d'exécution d'un algorithme de chiffrement peut également varier en raison de son activité mémoire, c'est-à-dire des accès au cache. Les échecs du cache augmentent notablement le temps d'exécution d'un algorithme au moment de l'exécution. Cette classe d'attaque temporelle ne manipule pas directement la mémoire cache.

V. Les mécanismes de sécurité :

Les techniques d'atténuation contre les attaques de timing et basées sur le cache peuvent être divisées en trois grandes classes selon les couche d'application applicable dans les infrastructures : application, système et matériel.

1. Niveau système

Au niveau du système nous aurons deux approches : [36, 9, 20]

1.1. Approches basées sur le système

L'atténuation au niveau du système d'exploitation dans les infrastructures cloud. Les attaques contre la déduplication de la mémoire dans les systèmes virtualisés ont conduit les industries du logiciel à modifier les paramètres système en désactivant ces fonctionnalités par défaut dans leurs systèmes, comme Amazon EC2. Ces approches peuvent être :

- **L'espacement du noyau :** fournit une protection concrète des pages partagées entre les machines virtuelles croisées en PaaS.
- **Le remplissage temporel :** de nettoyage du cache et de partitionnement dynamique. Le remplissage empêche ainsi un attaquant de mesurer le temps d'exécution de la fonction.
- **Le nettoyage du cache :** empêche d'obtenir l'état du cache après l'exécution de la fonction sensible.
- **Le partitionnement du cache :** permet de protéger les ressources d'un processus approuvé contre l'accès par un processus non approuvé pendant son exécution.
- **Le verrouillage de la ligne de cache :** permet le verrouillage et le multiplexage des lignes de cache pour chaque VM à l'aide d'une méthode logicielle. Le logiciel verrouille les pages d'une VM dans le cache partagé, un ensemble de lignes de cache étant verrouillé et attribué à chaque cœur de processeur. Lorsqu'une VM s'exécute sur un cœur, les pages de VM sont chargées sur les lignes verrouillées et ne peuvent pas être expulsées par une autre VM s'exécutant sur d'autres cœurs de processeur. Le multiplexage des lignes de cache permet de masquer les modèles d'accès au cache d'une machine virtuelle à partir de machines virtuelles co-résidentes.

1.2. Approches basées sur l'hyperviseur

Les contre-mesures appliquées à ce niveau pourraient être plus intéressantes pour l'atténuation des attaques par canaux auxiliaires que les mécanismes au niveau du système d'exploitation, car le fournisseur de cloud n'a pas besoin de modifier le système d'exploitation invité des clients. Ces approches sont :

- **Le planificateur :** permet empêcher les ACAs entre VM, un paramètre de planificateur dans Xen et KVM appelé ratelimit_us est utilisé pour interrompre une VM malveillante traçant une VM victime. Ce paramètre détermine la durée d'exécution minimale d'une CPU virtuelle (VCPU) sur une CPU physique. L'approche est simple à mettre en œuvre, mais peut dégrader l'efficacité de l'hyperviseur. Par exemple, le planificateur cloud peut être utilisé à cette fin. En fait, le planificateur peut décider au hasard d'allouer l'instance à un hôte de l'infrastructure conformément à la politique de placement d'instance pour éviter la co-résidence entre les VM malveillantes et victimes sur la même machine physique.
- **Le vidage du cache :** permet d'éliminer tout chevauchement d'accès au cache entre la VM victime et la VM adverse. Le vidage crée une forte isolation des données entre deux machines virtuelles, c'est-à-dire que chacune ne voit que ses propres données dans le cache. Cette approche côté serveur a notamment été implémentée dans Xen pour empêcher les ACA basés sur le cache Prime et Probe pour le cloud.
- **La coloration de page :** est une autre approche logicielle pour atténuer les ACA basées sur le cache. Une couleur spécifique est attribuée aux pages de chaque VM. Les pages avec la même couleur est ensuite mappée à un ensemble fixe de lignes de cache, uniquement accessible à la VM associée. La coloration des pages se fait de manière statique ou dynamiquement. La coloration de page statique dégrade les performances de l'environnement virtualisé et limite le nombre de machines virtuelles en cours d'exécution. Dans la coloration de page dynamique, les mécanismes de protection du cache ne sont actifs que lors de l'exécution d'opérations sensibles pour améliorer les performances.

2. Niveau Application

L'inefficacité des algorithmes cryptographiques pourraient conduire à des attaques par canal auxiliaire. La plupart des mécanismes de prévention existants pour les attaques par canal auxi-

liaire basées sur le cache sont basés sur des logiciels et sont associés à un système de chiffrement spécifique, comme OpenSSL, fournissent des bibliothèques partagées à plusieurs programmes simultanément pour économiser de la mémoire et mettre à jour les problèmes. Alors, cela peut conduire à des menaces potentielles sur le logiciel.

Pour éviter les attaques par canal auxiliaire sur AES, de nombreux types de mécanismes ont été proposés, comme les tables AES doivent être chargées dans le cache avant d'exécuter un cryptage afin que tous les accès à AES créent un hit de cache et aient donc un temps de cryptage constant. Pendant l'exécution d'AES, seules les opérations mathématiques doivent être utilisées à la place des recherches de table. vidage de la mémoire cache, partition de cache à l'aide de la coloration du cache et du masquage des adresses, construction d'une nouvelle implémentation de l'algorithme cryptographique qui résiste aux attaques par canal auxiliaire, limitation du canal auxiliaire basé sur le cache dans le cloud mutualisé à l'aide de la coloration de page dynamique. [36, 9]

3. Niveau matériel

L'atténuation au niveau du matériel fournit une isolation solide entre les unités de traitement. Les approches proposées incluent la modification de l'architecture du cache et l'intégration de nouvelles technologies matérielles, au niveau du cache ou en termes de traitement cryptographique car les ressources matérielles les plus ciblées sont les caches de processeurs, ce qui a conduit les industries de processeurs à apporter des modifications physiques aux microprocesseurs pour atténuer les attaques.

Sur la base des études menées dans les attaques de fuite d'informations contre le cache L1, il a été recommandé aux industries du processeur de désactiver le partage des caches entre le cœur et les threads pour éviter les expulsions de ligne de cache. Les processeurs modernes prennent en charge les configurations matérielles, tels que le multi-threading. Le multi-threading prend en charge la synchronisation du cache entre les threads du même cœur, ce qui a la caractéristique de base de certaines attaques de canaux auxiliaires, par laquelle la désactivation du multi-thread arrête l'attaque.[36, 9, 20]

Niveau	Cible	Solution	Limite
Système	Attaque basée sur le cache	L'ajout de bruit dans le cache permet d'entrer la commutation de contexte, Effacer les lignes du cache pour effacer les traces d'accès, Changer de contexte en fonction du nombre d'instructions	Non pris en charge par tous les processeurs, surcharge de performance très élevé
Application	Attaque basée sur le Timing	Écriture d'algorithmes à temps constant, Technique de randomisation pour masquer les données d'entrée, ajout de temps sous forme de bruit dans les fonctions, utilisation du compilateur pour éliminer les données et contrôler la dépendance du programme	Très difficile d'écrire tels programmes, implémentation complexe, dégradation des performances, surcharge de performance très élevé
Matériel	Attaque basée sur le cache	Attribution aléatoire des lignes de cache aux threads, Diviser le cache à différentes zones de manière dynamique, Diviser le cache en différentes zones de manière statique, Verrouiller les lignes de cache pendant l'exécution des threads	surcharge de Performance, Implémentation complexe

TABLE 2.1 – Tableau comparatif [9]

VI. Conclusion

Dans ce chapitre nous avons vu les concepts liés à l'attaque par canal auxiliaire tels que son fonctionnement, les techniques qu'elle utilise et enfin un mécanisme de sécurité que nous nous en servirons pour implémenter une solution contre ce type d'attaque qui peut avérer très sophistique dans un environnement cloud du point de vue des utilisateur du service cloud, ce qui peut entrainer la perte de confiance au cloud.

Chapitre 3

Analyse et Conception

I. Introduction

Les attaques par canal auxiliaire extraient des informations secrètes en surveillant le comportement du cache d'une victime. Normalement, cette attaque cible un cache L3, qui est partagé entre un espion et une victime. Par conséquent, un espion peut obtenir des informations secrètes sans alerter la victime. Pour résister à cette attaque, il faut une technique de détection robuste et performante.

Dans ce chapitre nous allons proposer une solution reposant sur la détection et l'atténuation contre cette attaque.

II. Étude des solutions existantes

Dans la mesure de l'amélioration constante du cache, des contre-mesures ont également été proposées telles que : le vidage du cache , la coloration de page, la technologie d'allocation du cache, etc. Nous allons juste décrire quelques une de ces contre-mesures.

1. Coloration de page

L'idée principale derrière la coloration de page est que le système d'exploitation contrôle l'affectation de ces bits de sorte que les blocs de mémoire appartenant à différents utilisateurs, processus ou VM ne se heurtent pas dans le cache. La coloration de page a été implémentée comme mécanisme pour éviter les fuites de cache. Le système d'exploitation ou hyperviseur colore les pages DRAM en attribuant une couleur différente à chaque combinaison de bits dif-

férente pour les bits qui sélectionnent la position d'un bloc de mémoire dans le cache. Lors des demandes d'allocation DRAM faites par les processus ou utilisateurs, le système d'exploitation attribue à chaque utilisateur uniquement des pages d'une même couleur et différentes à tout autre utilisateur du système, la coloration garantit que les pages appartenant à différents utilisateurs ne se heurtent pas dans le cache. Si les pages mémoire ne se heurtent pas, les attaques de type Prime et Probe ne sont pas possibles, car l'attaquant ne peut pas expulser les données des victimes, ni Flush and Reload.[13, 25, 29]

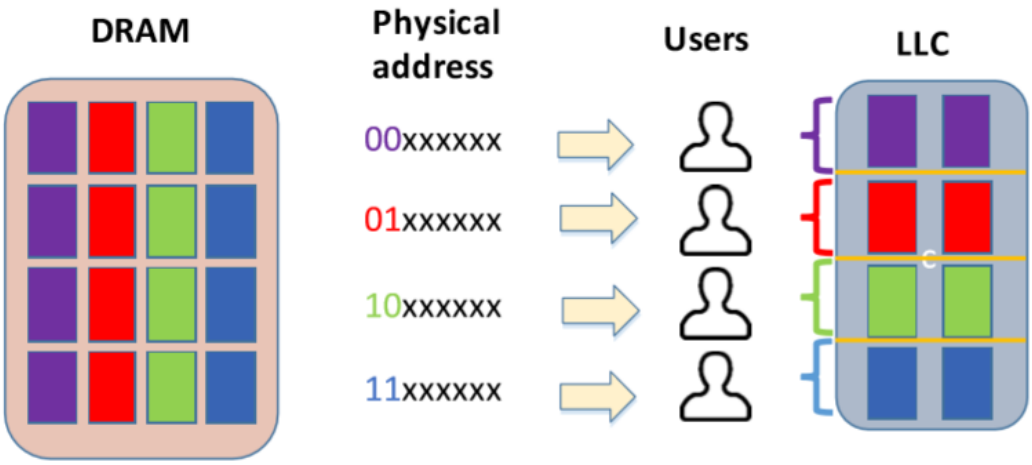

FIGURE 3.1 – Architecture de la coloration de page [25]

2. Cache Allocation Technologie

Intel a introduit la nouvelle technologie d'allocation de cache (CAT) dans ses processeurs pour améliorer les performances des applications sensibles à la latence en garantissant la capacité du cache aux applications prioritaires. CAT permet le partitionnement dynamique de la LLC entre différents cœurs de processeur. Différentes classes de service sont définies et peuvent être attribuées à chaque cœur. Ainsi, une partie de la LLC sera affectée à des cœurs sélectionnés, les lignes de cache dans chaque partie n'étant accessibles que par le cœur affecté. CAT a notamment été mis à profit pour atténuer les ACAs sur la LLC, la LLC a été divisée en partitions sécurisées et non sécurisées utilisant CAT. Ainsi, les pages de code sensible chargées dans les partitions sécurisées ne peuvent pas être expulsées par d'autres VM fonctionnant sur CPU.[9, 25]

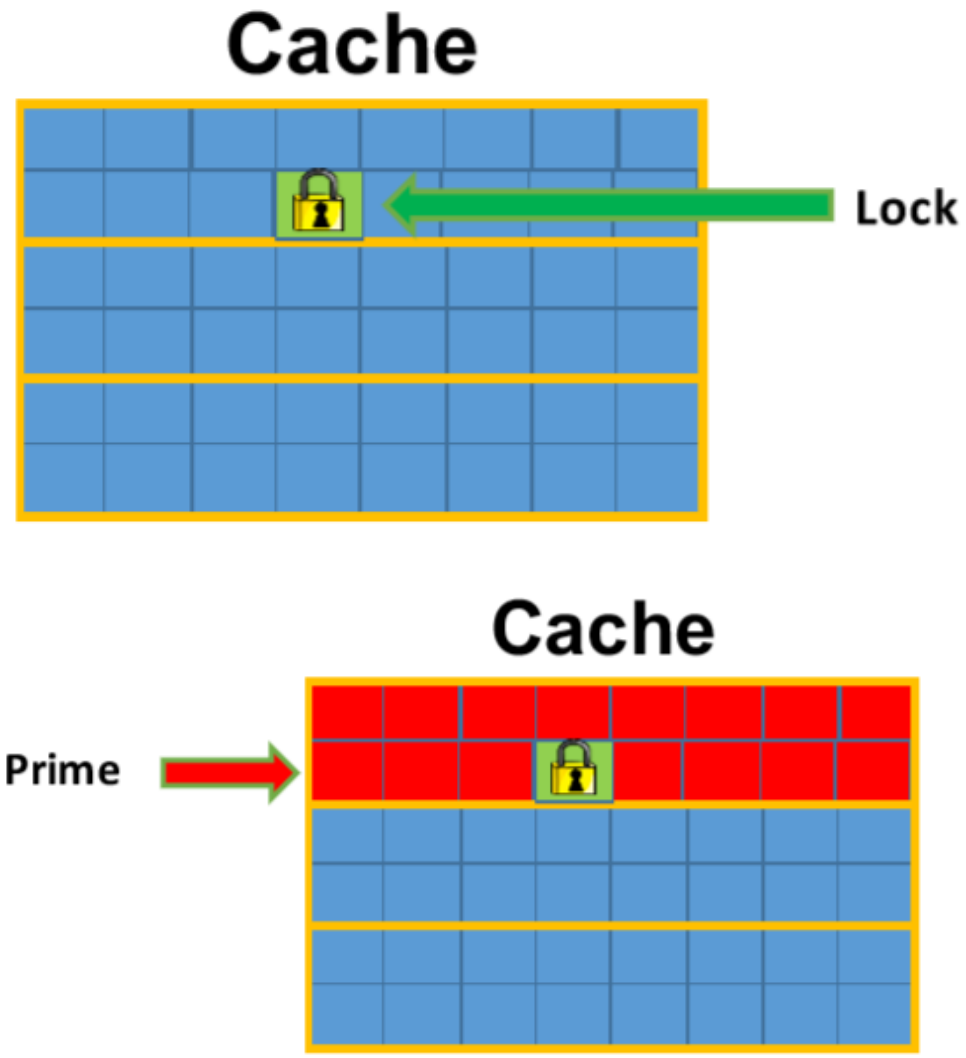

FIGURE 3.2 – Aperçu de CAT [25]

3. Vidage de cache

Permet d'éliminer et d'eviter tout chevauchement d'accès au cache entre la VM victime et la VM adverse. Le vidage crée une forte isolation des données entre deux machines virtuelles, c'est-à-dire que chacune ne voit que ses propres données dans le cache. Cette approche côté serveur a notamment été implémentée dans Xen pour empêcher les ACA basés sur le cache Prime et Probe pour le cloud.

III. Problématique

Dans l'étude des solutions existantes, nous constatons uniquement l'utilisation des techniques d'atténuations pour palier à ce problème. En outre, l'efficacité de ces techniques d'atténuation logicielles se fait au détriment des performances. Dans le souci de mettre les périphériques informatiques à l'abri de ces défauts matériels, les techniques de détection peuvent être utilisées comme première ligne de défense contre de telles attaques. L'utilisation de ces techniques d'atténuation coûteuses en termes de performances ne doivent pas être utilisé qu'après avoir détecté ces attaques. Pour que la stratégie de prévention basée sur la détection soit efficace, la détection doit être très précise, entraîner une surcharge système minimale au moment de l'exécution et être capable de détecter les attaques à un stade précoce, c'est-à-dire avant qu'elles ne soient terminées. Compte tenu de ces limitations des techniques d'atténuation logicielle, il est essentiel d'introduire une technique de détection pour ces attaques.

IV. Apprentissage profond(Deep learning)

1. Définition

Apprentissage profond est un sous-domaine de l'apprentissage machine, qui repose sur le traitement par les ordinateurs de grandes quantités de données à l'aide de réseaux de neurones artificiels dont la structure imite celle du cerveau humain. Chaque fois que de nouvelles informations sont intégrées, les connexions existantes entre les neurones sont susceptibles d'être modifiées et étendues, ce qui a pour effet de permettre au système d'apprendre les choses sans intervention humaine, de manière autonome, tout en améliorant la qualité de ses prises de décision et de ses prévisions.[40]

L'apprentissage profond (deep learning) est une notion issue du fait que les réseaux neurones disposaient de plus en plus de couches cachées et que le nombres élevés de couches devenait une source de problèmes. En effet, à partir d'un nombre de couches, le réseau neuronal n'était plus capable d'assimiler les informations et d'apprendre correctement.[45]

2. Approches d'apprentissage en profondeur

Les réseaux de neurones profonds réussissent dans l'apprentissage supervisé, l'apprentissage non supervisé, l'apprentissage par renforcement, ainsi que l'apprentissage hybride.[23, 24]

2.1. Apprentissage supervisé

Dans l'apprentissage supervisé, les variables d'entrée représentées par X sont mappées aux variables de sortie représentées par Y en utilisant un algorithme pour apprendre la fonction de mappage f.

$$Y = f(X)$$

Le but de l'algorithme d'apprentissage est d'approximer la fonction de mappage pour prédire la sortie (Y) pour une nouvelle entrée (X). L'erreur des prédictions faites pendant l'entraînement peut être utilisée pour corriger la sortie. L'apprentissage peut être arrêté lorsque toutes les entrées sont formées pour obtenir la sortie ciblée.

2.2. Apprentissage non supervisé

Dans l'apprentissage non supervisé, nous n'avons que les données d'entrée et aucune sortie correspondante à mapper. Cet apprentissage vise à apprendre les données en modélisant la distribution des données. Les algorithmes peuvent être en mesure de découvrir la structure présente dans les données. Les problèmes de clustering et les problèmes d'association utilisent l'apprentissage non supervisé.

2.3. Apprentissage par renforcement

L'apprentissage par renforcement utilise un système de récompense et de punition pour entraîner l'algorithme. En cela, l'algorithme ou un agent apprend de son environnement. L'agent obtient des récompenses pour les performances correctes et des pénalités pour les performances incorrectes. Par exemple, prenons le cas d'une voiture autonome, l'agent obtient une récompense pour avoir conduit en toute sécurité jusqu'à destination et une pénalité pour avoir quitté la route. De même, dans le cas d'un programme pour jouer aux échecs, l'état de récompense peut être de gagner la partie et la pénalité pour échec. L'agent essaie de maximiser la récompense et de minimiser la pénalité. Dans l'apprentissage par renforcement, on ne dit pas à l'algorithme comment effectuer l'apprentissage, il résout le problème seul .

2.4. Apprentissage hybride

L'apprentissage hybride fait référence à des architectures qui utilisent des composants génératifs (non supervisés) et discriminants (supervisés). La combinaison de différentes architec-

tures peut être utilisée pour concevoir un réseau neuronal hybride profond. Ils sont utilisés pour la reconnaissance des actions des humains à l'aide des fonctionnalités de banque d'actions et devraient produire de bien meilleurs résultats.

3. Les algorithmes du deep learning

Quelques algorithmes du deep learning :[47]

- Convolutional Neural Network (CNN)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory Networks (LSTMs)
- Stacked Auto-Encoders.
- Deep Boltzmann Machine (DBM)
- Deep Belief Networks (DBN)

4. Les réseaux de neurones

Les réseaux de neurones, communément appelés des réseaux de neurones artificiels sont des imitations simples des fonctions d'un neurone dans le cerveau humain pour résoudre des problématiques d'apprentissage de la machine (Machine Learning).[45]

Les réseaux de neurones peuvent être utilisés pour la régression ou la classification. Comme d'habitude dans l'apprentissage statistique, les paramètres sont estimés à partir d'un échantillon d'apprentissage. La fonction à minimiser n'est pas convexe, ce qui conduit à des minimiseurs locaux. Le succès de la méthode est venu d'un théorème d'approximation universel dû à Cybenko (1989) et Hornik (1991). [26]

4.1. Perceptron simple

Le perceptron simple est dit simple parce qu'il ne dispose que de deux couches ; la couche en entrée et la couche en sortie. Le réseau est déclenché par la réception d'une information en entrée. Le traitement de la donnée dans ce réseau se fait entre la couche d'entrée et la couche de sortie qui sont toutes reliées entre elles. Le réseau intégral ne dispose ainsi que d'une matrice de poids. Le fait de disposer d'une seule matrice de poids limite le perceptron simple à un classificateur linéaire.[45]

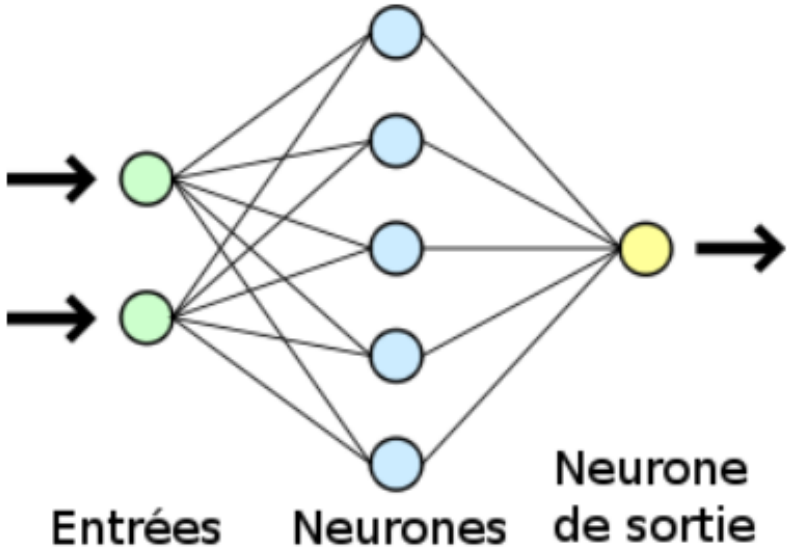

FIGURE 3.3 – Perceptron simple [45]

4.2. Perceptron multicouche

Un perceptron multicouche (ou réseau de neurones) est une structure composée de plusieurs couches cachées de neurones où la sortie d'un neurone d'une couche devient l'entrée d'un neurone de la couche suivante. De plus, la sortie d'un neurone peut également être l'entrée d'un neurone de la même couche ou d'un neurone de couches précédentes (c'est le cas des réseaux de neurones récurrents). Sur la dernière couche, appelée couche de sortie, une fonction d'activation sera appliqué selon le type de problèmes : régression ou classification.[26, 45]

Un perceptron multicouche est donc mieux adapté pour traiter les types de fonctions non-linéaires.

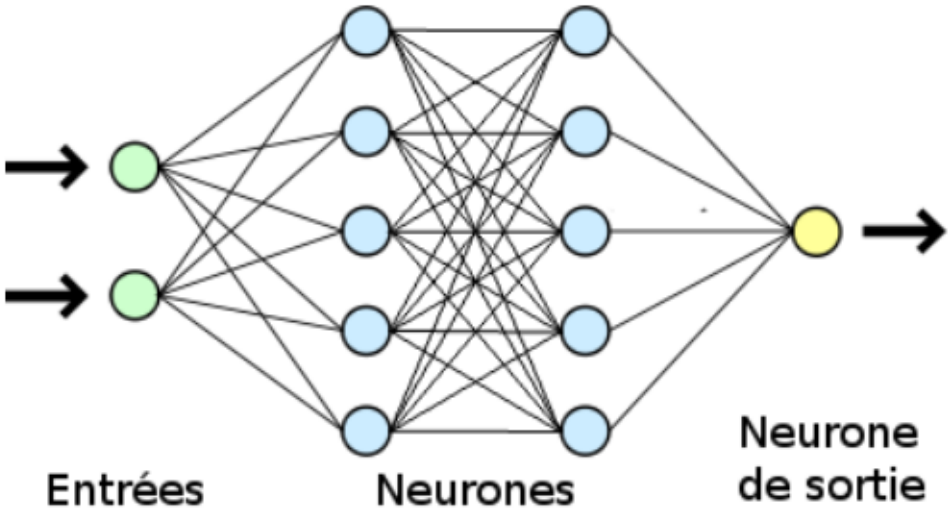

FIGURE 3.4 – Perceptron multicouche [45]

5. Domaine d'applications du Deep Learning

Les réseaux d'apprentissage en profondeur peuvent être utilisés dans une variété d'applications qui sont :[24]

- les voitures autonomes
- reconnaissance automatique d'image
- l'assistant virtuel de Google
- la reconnaissance visuelle
- la détection de fraude
- la détection d'attaque
- les soins de santé
- la traduction automatique
- prévision de taux de marché
- prévision de tremblement de terre

V. Description de la solution proposée

Notre solution consiste à mettre en addition un système de détection pour combler les techniques d'atténuation de l'attaque par canal auxiliaire.Ce système est divisé en deux services : Le service de détection et le service d'atténuation.

1. Service de détection

Notre technique de détection proposée effectue la détection des attaques par canal auxiliaire après avoir observé une variation des compteurs CPU. Pour cela, Intel PCM (Performance Counter Monitor) et des algorithmes d'apprentissage profond sont utilisés pour mesurer la valeur des compteurs CPU.

1.1. Phase de surveillance

Cette phase consiste à veiller sur le fonctionnement du cache en cas de variation. Cette phase nous permettra de visualiser l'état du cache par le biais d'un outil appelé intel PCM. Intel PCM permet la mesure de l'état du cache lors de l'attaque en temps réel. Lors du monitoring avec l'outil intel PCM les paramètres suivants sont utilisés : IPC, FREQ, AFREQ, L3MISS, L2MISS, L3HIT, INST, ACYC.

- IPC : instructions par cycle CPU
- FREQ : relation à la fréquence nominale du processeur
- AFREQ : relation à la fréquence nominale du processeur en état actif
- L3MISS : échec du cache L3
- L2MISS : échec du cache L2
- L3HIT : taux de succès du cache L3 (0,00-1,00)
- INST : instruction
- ACYC : cycle en état actif

1.2. Phase de l'apprentissage profond(Deep learning)

Lorsque la détection d'anomalies est mise en œuvre par des techniques d'apprentissage profond, des algorithmes sont utilisés afin de réaliser une différenciation entre un comportement anormal et normal, sur la base de l'observations précédentes. Elle est appelée prédiction.

L'algorithme d'apprentissage profond fait une prédiction quant à la catégorie d'une nouvelle observation, sur la base de l'observation précédentes déjà traitées. La catégorie d'une observation, par exemple la catégorie de comportement normal et anomalie, est assignée à une observation grâce à une étiquette, aussi appelée label. Les différents labels indiquent alors des classes d'appartenance. Les observations labellisées permettant de construire le modèle sont appelées données d'entrainement(data set).

Le modèle créé fait des prédictions quant à la classe de tout nouveau cas non connu (appelé données de test) et ne faisant donc pas partie du jeu de données d'entrainement.

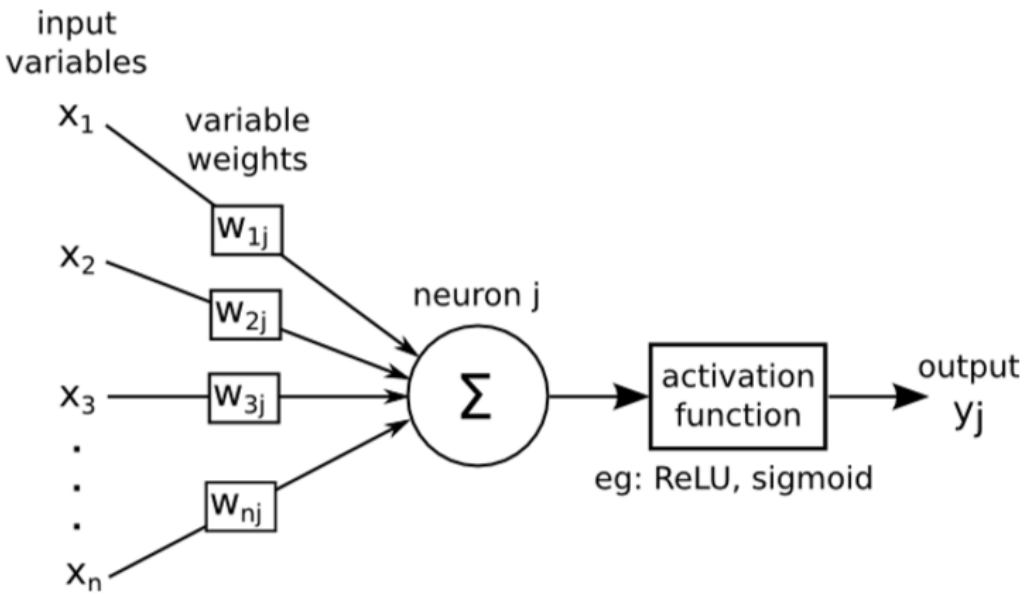

FIGURE 3.5 – Répresentation d'un neurone artificiel [26]

Les variables $x = (x_1, ..., x_n)$ représentent les paramètres en entrée du neurone, $w = (w_1, ..., w_n)$ représentent les vecteurs de poids, y : la sortie du neurone et b le biais du neurone.

En entrée du neurone, nous avons la combinaison suivante : $S(x) = b + \sum_{i=1}^{n}(w_i, x_i)$ et b est appelé le biais du neurone. En applicant la fonction d'activation sigmoïde, nous obtenons ceci en sortie du neurone de la couche de sortie :

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

La Figure 3.5 représente notre réseau de neurone avec des paramètres en entrée du neurone cité dans la Phase de surveillance et en sortie le paramètre BIN qui vaut 1 s'il y a une attaque et

0 pour un utilisation normale sans attaque.

2. Service de l'atténuation

Notre solution implique l'intervention de notre technique de vidage du cache dans un système Cloud. Vider un cache de haut niveau sur une machine moderne peut prendre du temps. Donc, il est préférable de vider le cache uniquement lorsque cela est nécessaire.

VI. L'architecture utilisée

Dans cette section, nous décrivons à l'aide d'un schemas l'architure de notre solution.

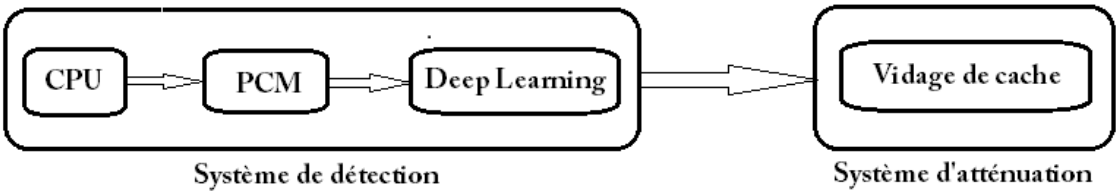

FIGURE 3.6 – Architecture du système

VII. L'algorithmes

Dans cette section, nous allons détailler le principe de notre système pour mieux comprendre son fonctionnement.

Algorithme général :

Entrée : Donnée d'entrainement

Sortie : résultat du teste

Donnée = Exécution de PCM avec sauvegarde au format csv

Modèle = fichier du modèle de deep learning

résultat_test = Exécution du modèle avec donnée

Si résultat_test == 1 alors

affiche(Utilisation anormale)

Sinon

affiche(Utilisation normale)

Finsi

Algorithme utilisé pour le deep learning :

Algorithme :

Entrée : Data_set, Data_set_test

Sortie : résultat de la prédiction

Donnée = lecture du Data_set

Modèle = création du modèle avec les paramètres de Donnée

compilation du modèle avec les pertes, l'optimiseur et la metrique

affiche des statistiques du modèle et le résulat de l'entrainement

Donnée_test = lecture des Data_set_test

prédiction = prédiction à travers le modèle avec les paramères de Donnée_test

résultat_prédiction = Exécution de la prédiction

affiche(résultat_prédiction)

VIII. Conception

La partie conception de notre système nous permet de modéliser et décrire différents besoins du système. Nous allons utiliser deux(2) diagrammes : diagramme de cas d'utilisation et diagramme de classe.[44]

1. UML(Unified Modeling Language)

C'est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter, spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

2. Diagramme UML

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle.

2.1. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils représentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

Notre diagramme de cas d'utilisation est défini comme suit :

- **Acteur :**User
- **Cas d'utilisation sont :** Gérer PCM, Exécuter modèle, Vider cache, Alerter user, Générer csv, Monitorer cache.

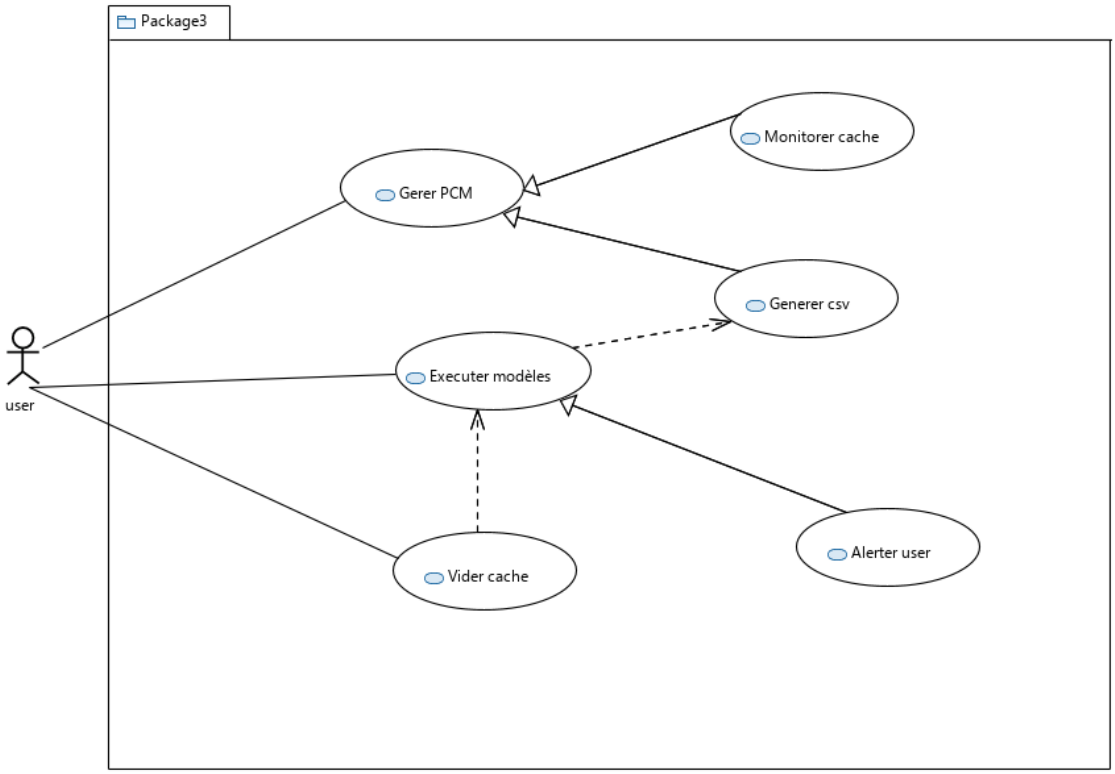

FIGURE 3.7 – Diagramme de cas d'utilisation

2.2. Diagramme de classe

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ses classes. Dans notre cas, il contient sept(7)classes à savoir : User, Attaque, Modèle Deep Learning, PCM, csv, Alerte, Vidage.

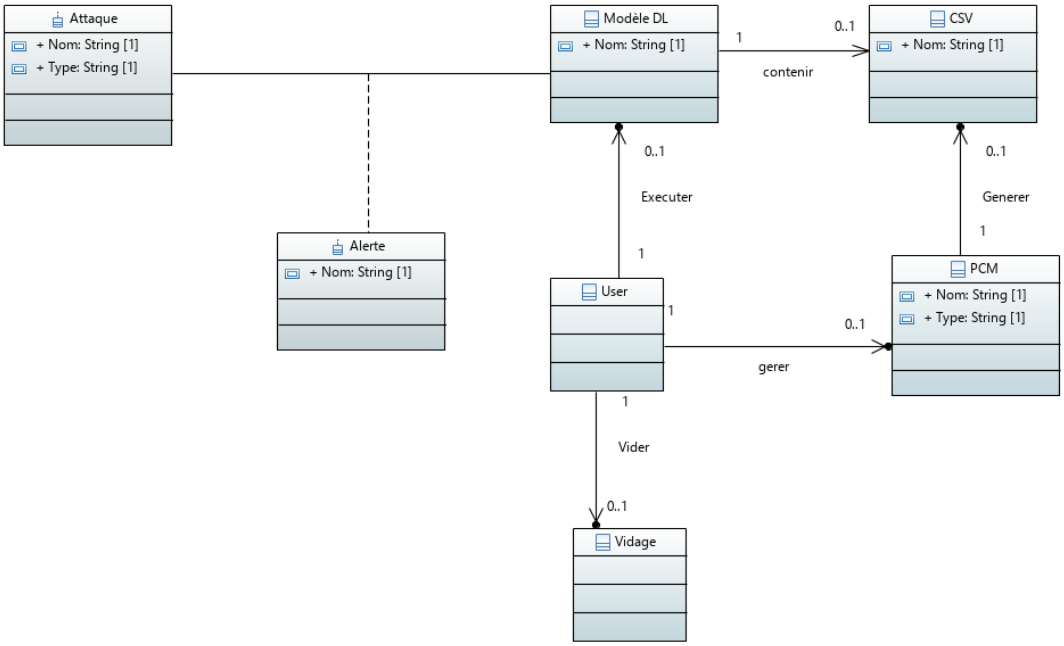

FIGURE 3.8 – Diagramme de classe

IX. Conclusion

Il y'a plusieurs solution pour contrer l'attaque par canal auxiliaire, chacune d'elle présente des inconvénients et des avantages en terme de ressource et de performance. Dans ce chapitre nous avons décris les éléments clés de notre solution proposée c'est à dire comment fonctionne notre système à savoir la détection de l'attaque par canal auxiliaire et son élimination.

Chapitre 4

Mise en oeuvre de la solution proposée

I. Introduction

Dans ce chapitre nous présentons l'objectif du projet qui est de mener à terme une solution efficace. De ce fait, nous décrivons l'implémentation de notre solution et l'environnement de travail et on terminera par le résultat du teste.

II. Environnement Matériel utilisé

- Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40GHz
- 12 Go de RAM
- Disque dur 1T HDD
- Système d'exploitation 64 bits, processeur x64.

III. Environnement Logiciel utilisés

Le cloud computing se repose sur la virtualisation. Ce pendant, nous avons réalisé ce projet en utilisant le logiciel VMware workstation qui permet le virtualisation dans le but de pouvoir simuler un environnement cloud.

- VMware workstations
- Kali linux
- Google Colab

- Les framework du deep learning
- Intel PCM

1. Présentation de VMware

VMWare permet de simuler un ou plusieurs PC sur un ordinateur Linux ou Windows. VMWare émule un véritable PC dont tous ses périphériques virtuels sont indépendants de ceux de l'hôte.

L'éditeur VMWare propose plusieurs types de logiciels de virtualisation destinés chacun à différents usages. VMWare workstation (logiciel à installer sous Windows ou Linux) est destiné à être installé sur des ordinateurs de bureau tandis que VMWare ESX est lui plutôt développé pour les architectures serveurs. VMWare ESX est en fait un système d'exploitation basé sur un noyau Linux et dédié à l'hébergement de serveurs virtuels. Cette architecture permet une sécurité accrue de la machine hôte.

Le principe de fonctionnement de VMWare Workstation est de créer des machines virtuelles dans leur totalité. Il est nécessaire d'installer un système d'exploitation hôte normal afin d'installer la couche logicielle nécessaire pour émuler le matériel.

Tout type de système d'exploitation peut être le système hôte de machines virtuelles grâce à VMWare (Windows, Linux ...) et il est possible d'installer n'importe quel OS sous VMWare. Cette solution est avantageuse dans le but d'avoir accès aux logiciels fonctionnant sous windows alors que Linux est le système natif par exemple.

Dans le cas de VMWare ESX, l'OS hôte est simplement plus spécifique qu'une version de Windows ou Linux normale mais l'architecture est la même que celle de la version Workstation. Il est indispensable d'installer un nouveau système d'exploitation pour chaque nouvelle machine virtuelle à créer. La couche matérielle ajoutée à l'OS hôte est un des facteurs rendant les performances des machines virtuelles moins bonnes que celles du système d'exploitation natif.[46, 43]

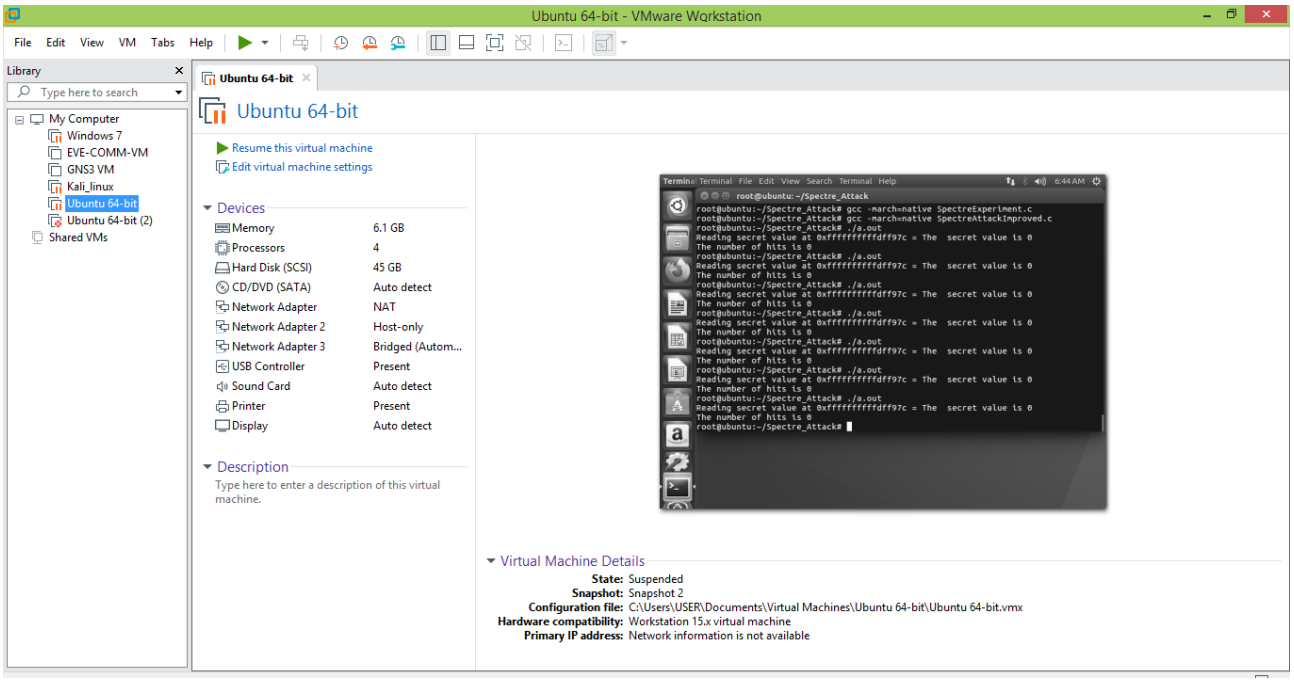

FIGURE 4.1 – L'interface d'utilisation de vmware workstation

2. Présentation de Kali linux

Kali Linux est une distribution Linux sortie le 13 mars 2013, basée sur Debian. La distribution a pris la succession de Backtrack, qui était basé sur Slackware jusqu'à la version 3 puis Ubuntu depuis la version 4. Backtrack était issu de la fusion de Whax et Auditor. L'objectif de Kali Linux est de fournir une distribution regroupant l'ensemble des outils nécessaires aux tests de sécurité d'un système d'information, notamment le test d'intrusion.

Cette distribution est utilisée par des auditeurs en sécurité des systèmes d'information dans le cadre de l'audit automatisé de premier niveau, permettant d'évaluer la sécurité intrinsèque d'un environnement. Il est aussi utilisé par des hackers.[42]

3. Intel PCM

Performance Counter Monitoring (PCM) est un outil qui permet aux utilisateurs de surveiller les valeurs de compteur de performances du cœur du processeur. PCM aide les utilisateurs à surveiller le taux de changement du compteur interne (par exemple, instruction par cycle, échec du cache L1, L2 et L3) de chaque CPU en temps réel.

Intel PCM peut être exécuté en tant que fichier binaire en effectuant une compilation. Intel

PCM a une option CSV qui imprime plusieurs valeurs de compteur des CPU dans un fichier CSV pendant une durée définie. Intel PCM peut être utilisé pour distinguer les caractéristiques des attaques de canal auxilliaire via un changement de compteurs.[19]

4. Frameworks d'apprentissage profond(Deep learning)

Chaque framework du deep learning aide à modéliser un réseau plus rapidement sans entrer dans les détails des algorithmes.[23, 24]

1. **TensorFlow :** TensorFlow, développé par Google brain, prend en charge des langages tels que Python, C ++ et R. Il nous permet de déployer nos modèles d'apprentissage en profondeur dans les processeurs ainsi que les GPU .
2. **Keras :** Keras est une API, écrite en Python et exécutée sur TensorFlow. Cela permet une expérimentation rapide. Il prend en charge à la fois les CNN et les RNN et fonctionne sur des CPU et des GPU.
3. **PyTorch :** PyTorch peut être utilisé pour créer des réseaux de neurones profonds ainsi que pour exécuter des calculs tensoriels. PyTorch est un package basé sur Python qui fournit des calculs Tensor. PyTorch fournit un cadre pour créer des graphes de calcul.
4. **Caffe :** il est également open source. Caffe se distingue des autres frameworks par sa rapidité de traitement ainsi que par l'apprentissage des images. Le framework Caffe Model Zoo nous permet d'accéder à des modèles pré-entraînés, ce qui nous permet de résoudre divers problèmes sans effort.
5. **Deeplearning4j :** Deeplearnig4j est implémenté en Java, et par conséquent, il est plus efficace que Python. La bibliothèque de tenseurs ND4J utilisée par Deeplearning4j offre la possibilité de travailler avec des tableaux ou des tenseurs multidimensionnels. Ce cadre prend en charge les processeurs et les GPU. Deeplearnig4j fonctionne avec des images, csv ainsi que du texte en clair.

5. Présentation de Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Deep Learning directement dans le cloud.

Sans donc avoir besoin d'installer quoi que ce soit sur l'ordinateur à l'exception d'un navigateur. [41]

Colab nous permet de tirer pleinement parti des bibliothèques populaires Python pour analyser et visualiser des données parmi ces bibliothèques on a numpy pour générer des données aléatoires et matplotlib pour les visualiser.

On peut importer des données dans les notebooks Colab depuis le compte Google Drive, y compris depuis des feuilles de calcul, ainsi que depuis GitHub et de nombreuses autres sources.

Colab permet :

- d'améliorer vos compétences de codage en langage de programmation Python et R ;
- d'écrire et d'exécuter du code Python dans votre navigateur
- de développer des applications en Deep Learning en utilisant des bibliothèques Python populaires telles que Keras, TensorFlow, PyTorch et OpenCV ;

Il offre les avantages suivants :

- Aucune configuration requise.
- Accès gratuit aux GPU.
- Partage facile

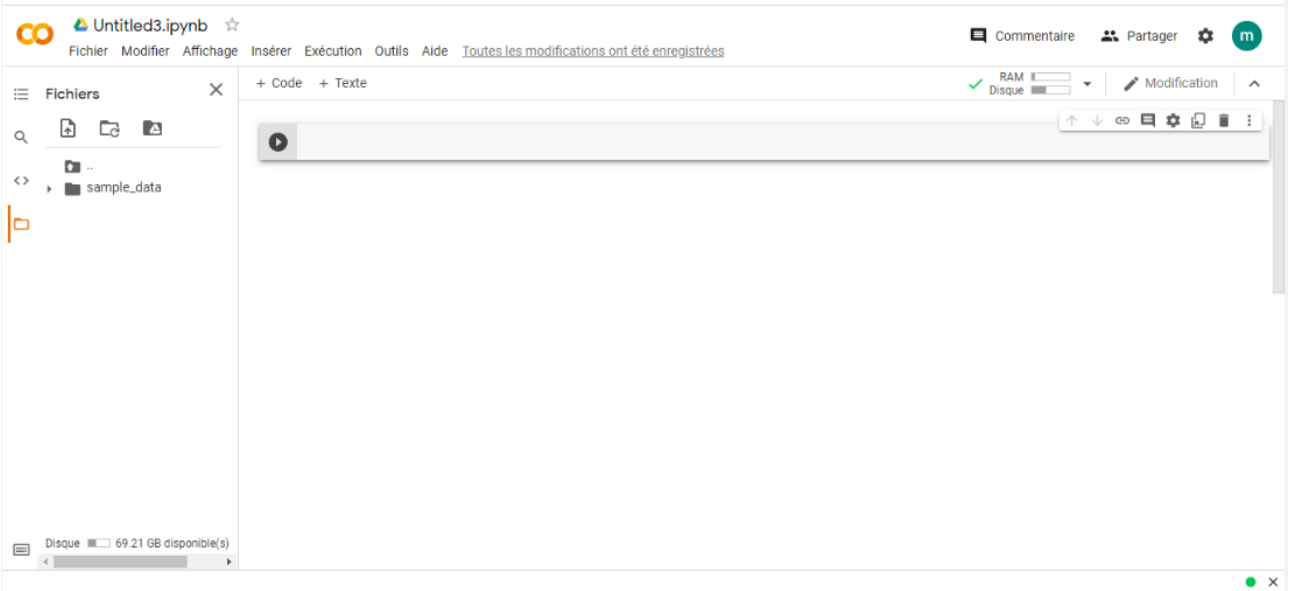

FIGURE 4.2 – L'interface d'utilisation du plateforme google colab

IV. Phase d'implémentation

1. Phase de monitoring avec PCM

Les algorithmes d'apprentissage automatique détectent les anomalies en fonction de l'état du cache mesuré. La phase de monitoring de notre solution consiste en l'utilisation de l'outil PCM pour nous permettre d'extraire des information au format csv concernant la cache.

La commande ./pcm.x permet surveiller l'état du cache avec l'option -csv pour générer le fichier csv de l'état du cache.

FIGURE 4.3 – Commande pour le monitoring du cache

On a le résultat du monitoring sous forme de fichier au format csv. Lors d'une utilisation normale sans attaque les valeurs des paramètres sont :

- L3MISS < 1
- L2MISS < 1
- 150 > INST > 1
- 300 > ACYC > 10
- 0.5 > AFREQ > 0.1

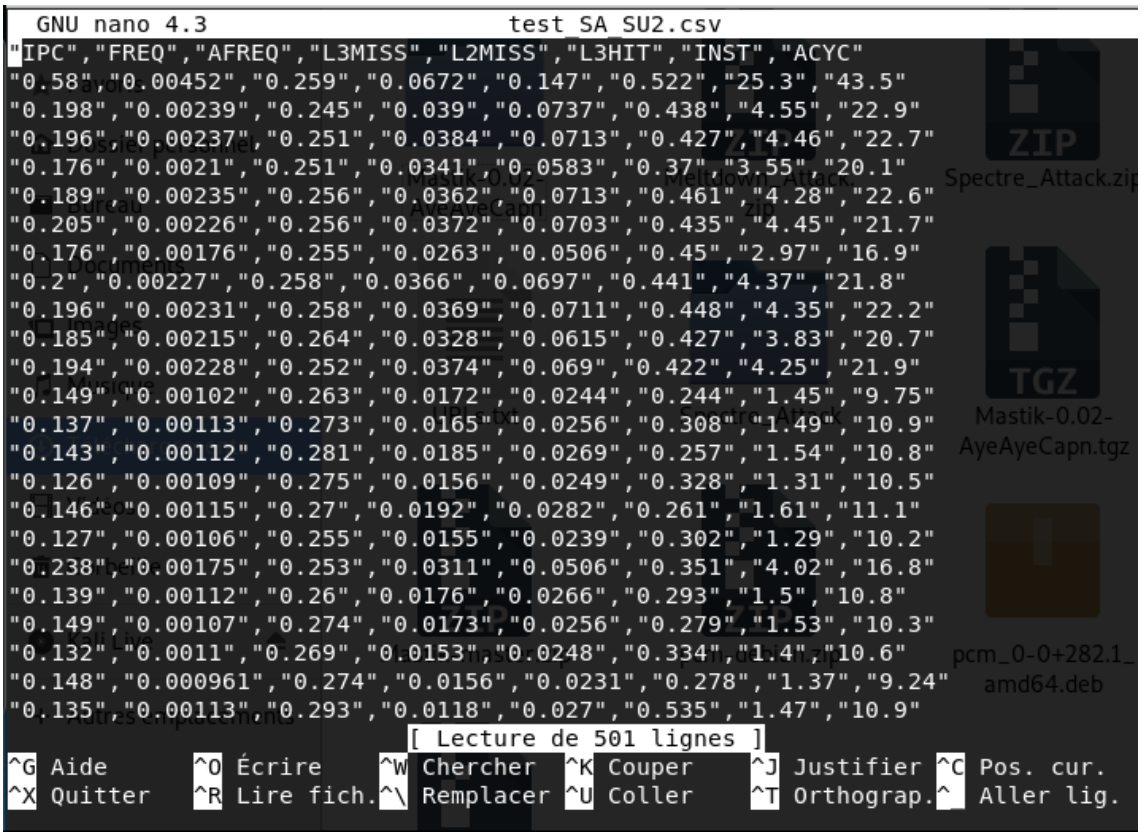

FIGURE 4.4 – Extrait du résultat d'une utilisation normale sans attaque

De même pour une attaque.

- L3MISS > 1
- L2MISS > 1
- 1000 > INST > 100
- 1500 > ACYC > 500
- 1 > AFREQ > 0.5

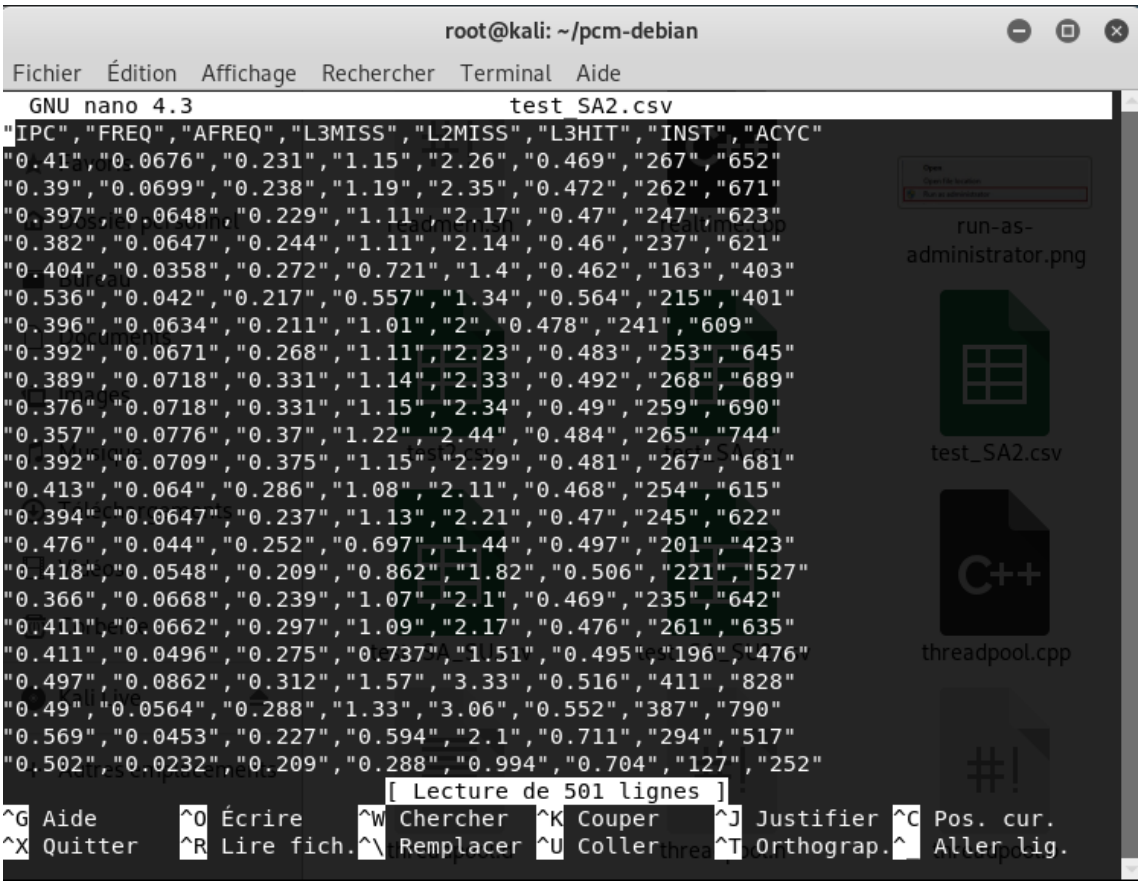

FIGURE 4.5 – Extrait du résultat d'une attaque

2. Phase de détection avec Deep learning

Dans cette phase nous avons utilisé deep learning pour la détection en faisant la prédiction d'une utilisation normale ou anormale.

2.1. Conception du modèle d'apprentissage

Ceci est une illustration de notre modèle d'apprentissage pour la détection.

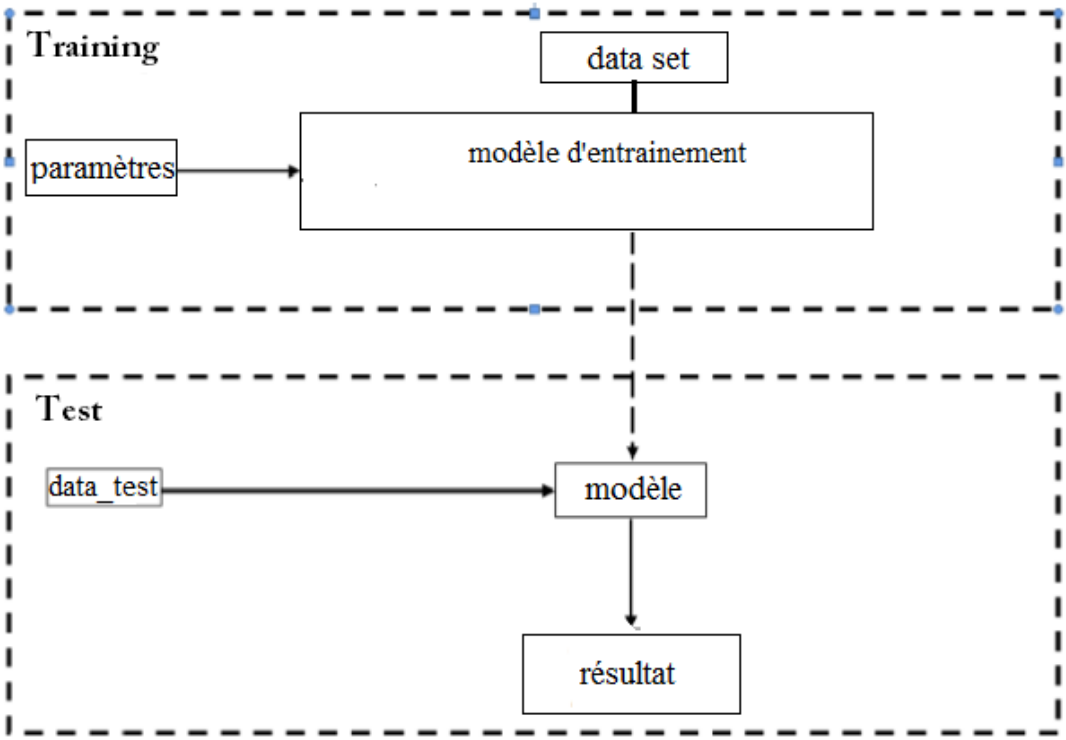

FIGURE 4.6 – Modèle d'apprentissage

2.2. Préparation de donnée

Nous avons utilisé un jeu de donnée d'entrainement combinant les différents usages générés par l'outil PCM.

	IPC	FREQ	AFREQ	L3MISS	L2MISS	L3HIT	INST	ACYC	BIN
0	0.410	0.06760	0.231	1.1500	2.2600	0.469	267.00	652.0	1
1	0.390	0.06990	0.238	1.1900	2.3500	0.472	262.00	671.0	1
2	0.397	0.06480	0.229	1.1100	2.1700	0.470	247.00	623.0	1
3	0.382	0.06470	0.244	1.1100	2.1400	0.460	237.00	621.0	1
4	0.404	0.03580	0.272	0.7210	1.4000	0.462	163.00	403.0	1
...	...	...	...	...	...	...	...	...	...
995	0.130	0.00112	0.320	0.0171	0.0253	0.276	1.39	10.8	0
996	0.110	0.00151	0.315	0.0197	0.0298	0.294	1.61	14.5	0
997	0.441	0.00217	0.292	0.0318	0.0539	0.379	9.18	20.8	0
998	0.372	0.00397	0.288	0.0514	0.1150	0.532	14.20	38.1	0
999	0.138	0.00133	0.276	0.0206	0.0313	0.289	1.77	12.8	0

FIGURE 4.7 – Aperçu du jeu de donnée d'entrainement

- IPC : instructions par cycle CPU
- FREQ : relation à la fréquence nominale du processeur
- AFREQ : relation à la fréquence nominale du processeur en état actif
- L3MISS : échec du cache L3
- L2MISS : échec du cache L2
- L3HIT : taux de succès du cache L3 (0,00-1,00)
- INST : instruction
- ACYC : cycle en état actif

2.3. Code source de l'implémentation

```
from keras.models import Sequential
import pandas as pd
```

```
from sklearn import preprocessing
from keras.layers import LSTM, Dense,Dropout
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv('Classeur2.csv')
dataset = df.values
X = dataset[ :, 0 :8]
Y = dataset[ :, 8]
model = Sequential()
model.add(Dense(32, activation='relu', input_dim=8))
model.add(Dropout(rate=0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
hist = model.fit(X, Y, epochs=1000, batch_size=25)
score = model.evaluate(X, Y)
print("Le taux de réussite est de : %.2f%%" % (score[1] * 100))
test_df = pd.read_csv("test.csv")
data_test = test_df.values
X2 = data_test[ :, 0 :8]
prediction = model.predict(X2)
rounded = [round(x[0]) for x in prediction]
a = 0
b = 0
for i in rounded :
if rounded[i] == 0 :
a = a + 1
else :
b = b + 1
if a < b :
etat = 'Utilisation anormale'
else :
```

etat = 'Utilisation normale'

2.4. Modèle d'apprentissage

Notre modèle d'entrainement contient :

- Deux couches Dense avec 32 unité en entrée et une seule unité en sortie.
- Une seul couche Dropout pour overfitting
- Paramètre d'entré 8 et sortie 1.
- loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']
- epochs=1000, batch_size=25
- Fonction d'activation : Sigmoid

2.5. Résultat de l'entrainement

Lors de la compilation du modèle, plus la valeur de l'epochs augmente plus on a une bonne précision.

```
Epoch 990/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0489 - accuracy: 0.9733
Epoch 991/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0553 - accuracy: 0.9712
Epoch 992/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0722 - accuracy: 0.9666
Epoch 993/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0502 - accuracy: 0.9730
Epoch 994/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0475 - accuracy: 0.9812
Epoch 995/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0484 - accuracy: 0.9777
Epoch 996/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0545 - accuracy: 0.9761
Epoch 997/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0595 - accuracy: 0.9679
Epoch 998/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0531 - accuracy: 0.9842
Epoch 999/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0628 - accuracy: 0.9660
Epoch 1000/1000
40/40 [==============================] - 0s 2ms/step - loss: 0.0529 - accuracy: 0.9699
32/32 [==============================] - 0s 1ms/step - loss: 0.0262 - accuracy: 0.9870
Le taux de réussite est de : 98.70%
```

FIGURE 4.8 – Le résultat de la compilation du modèle

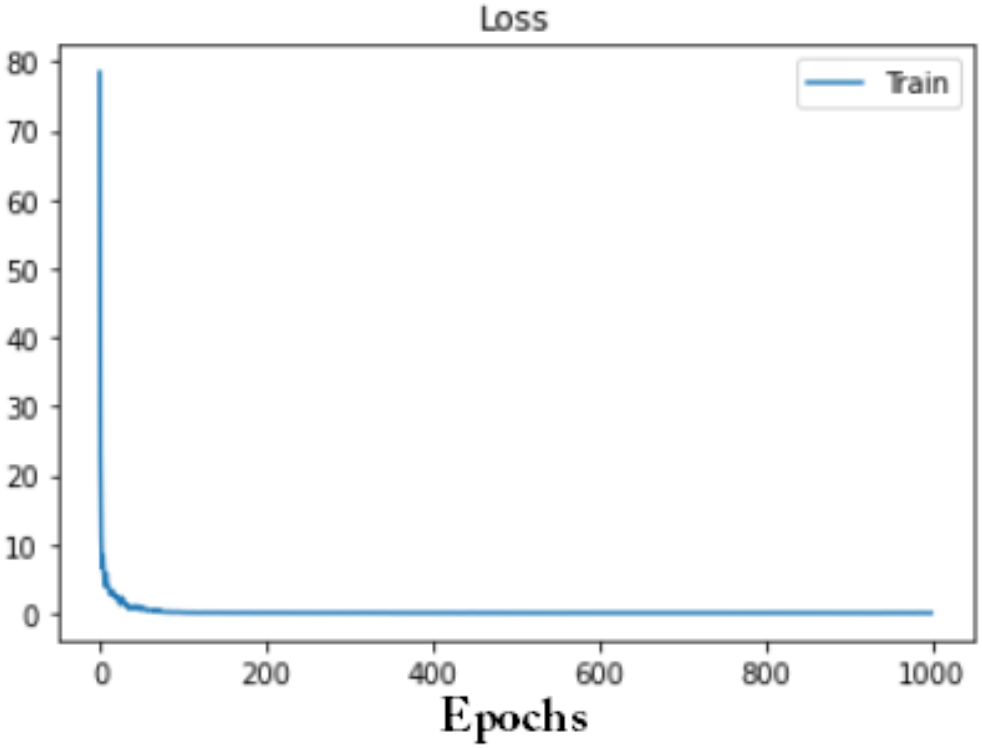

FIGURE 4.9 – Graphe de perte lors de l'entrainement

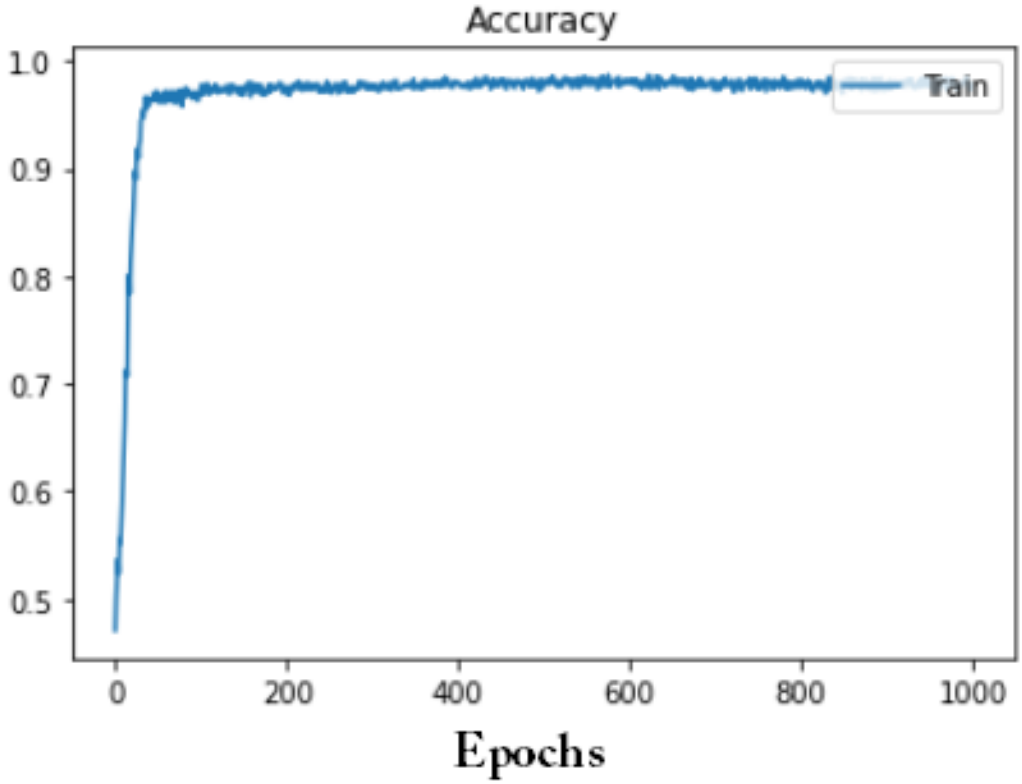

FIGURE 4.10 – Graphe de précision lors de l'entrainement

2.6. Résultat du teste

Ainsi sont les résultats de notre fameux teste.

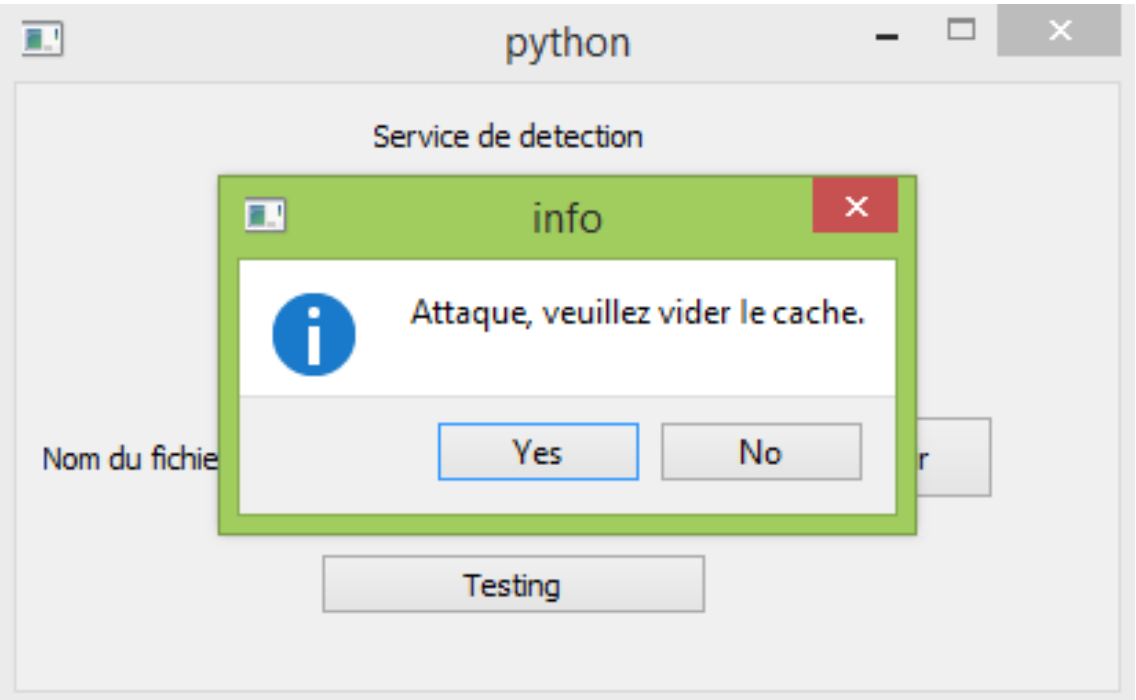

FIGURE 4.11 – Alerte d'une utilisation anormale(attaque)

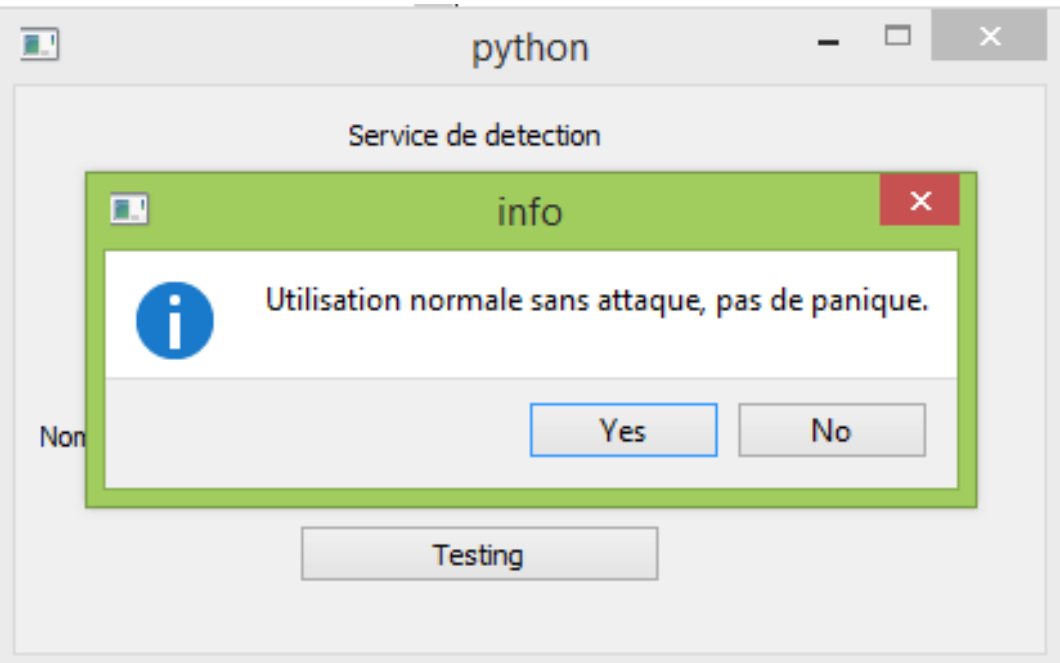

FIGURE 4.12 – Alerte d'une utilisation normale(Sans attaque)

3. Phase de l'atténuation

Dans cette phase, nous avons utilisé la commande ci-dessous pour vider le cache de ses débris.

$ free -h : permet de visualiser la mémoire cache comme indiquer sur la figure.

$ vm.drop=3 : permet de vider le cache.

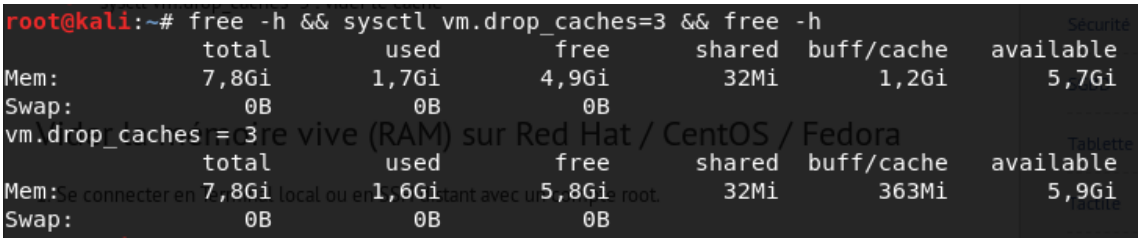

FIGURE 4.13 – Aperçu du jeu de donnée d'entrainement

V. Conclusion

Après avoir décrire les différentes étapes de la solution proposée telle que la détection et l'atténuation. Dans ce chapitre, nous passons à la réalisation de ces différentes étapes : le monitoring du cache , l'apprentissage profond et le vidage du cache en cas d'attaque.

Conclusion et Prespective

En Conclusion, ce travail consistait à décrire le cloud computing dans sa globalité et les attaques aux quelles font face cette technologie surtout l'attaque par canal auxiliaire et la mise en oeuvre d'une solution pour empêcher cette attaque.

Cependant, il nécessite une étude comparative des mécanismes de sécurité existants et d'adopter une solution plus robuste afin d'en servir pour pouvoir proposer une meilleure solution en terme de complexité, et de performance c'est-à-dire qui n'est pas gourmand en terme de mémoire et plus rapide.

Nous avons adopté une technique de détection en addition des mécanismes de sécurité pour donner plus de précision et certitude dans nos résultats. Cette technique permet aux utlisateurs d'appliquer les mécanismes de sécurité existants sauf si cela est nécessaire et elle éfficace en terme de performance.

Par concéquent, la méthode proposée sera vulnérable si un attaquant a un accès privilégié aux ressources (par exemple, le fichier CSV 4.7) pour exécuter le programme de détection.

Dans nos futurs travaux, nous aborderons ces problèmes restants et continuerons à améliorer la technique proposée afin qu'elle puisse détecter d'autres techniques d'attaques de canal auxiliaire dans un environnement cloud computing.

Bibliographie

Articles et Publications

[1] Peter Mell et al, The NIST définition of cloud computing, National Institute of Standards and Technology, Special Publication 800-145, 2011.

[2] Hassan EL Alloussi et al, Etat de l'art de la sécurité dans le Cloud Computing, Novembre 2012.

[3] Dr. Zertal Soumia, Cours sur le cloud et la virtualisation, Université Larbi Ben M'hidi-Oum El Bouaghi, Edition 2.0, 2019/2020 .

[4] Cloud Security Alliance : Security Guidance for Critical Areas of Focus in Cloud Computing v3.0, 2011.

[5] Top Threads Working Group, et al. The notorious nine : cloud computing top threats in 2013. Cloud Security Alliance, 2013.

[6] YU et Shui, Distributed Denial of Service Attack and Defense, Springer New York, 2014.

[7] Dan Hubbard et al, Top Threats to Cloud Computing V1.0, Cloud Security Alliance, 2010.

[8] Navjot Singh Brar et al, Study of Virtual Side Channel Attack in Cloud Computing, IJEDR, 2015, Volume 3, Issue 3.

[9] Mohammad-Mahdi Bazm et al, Side Channels in the Cloud : Isolation Challenges, Attacks, and Countermeasures, 2017.

[10] Bhrugu Sevak, Security against Side Channel Attack in Cloud Computing, IJEAT, Volume 2, Issue 2, 2012.

[11] Younis A.Younis et al, Cache Side-Channel Attacks in Cloud Computing, 2014.

[12] Guillaume Duc, Les attaques par canaux auxiliaires : un danger encore méconnu, Télécom ParisTech, 2016.

[13] Michael Godfrey et al, Preventing Cache-Based Side-Channel Attacks in a Cloud Environment ; IEEE Transactions on Cloud Computing, 2014.

[14] Maria Mushtaq et al, Machine Learning For Security : The Case of Side-Channel Attack Detection at Run-time, IEEE, 2018.

[15] Fangfei Liu et al, CATalyst : Defeating Last-Level Cache Side Channel Attacks in Cloud Computing, IEEE, 2016.

[16] Zecheng He et al, How secure is your cache against side-channel attacks, 2017.

[17] Tianwei Zhang et al, CloudRadar : A Real-Time Side-Channel Attack Detection System in Clouds, 2016.

[18] Yinqian Zhang et al, Cross-VM Side Channels and Their Use to Extract Private Keys, 2012.

[19] Jonghyeon Cho et al, Real-Time Detection for Cache Side Channel Attack using Performance Counter Monitor, Applied Science, 2020.

[20] Ajay Kaarthic J et al, A Survey on Cross VM Side Channel Attack In Cloud Computing, International Journal of Research and Analytical Reviews (IJRAR), 2018, Volume 5, Issue 3.

[21] Reza Montasari et al, Are Timing-Based Side-Channel Attacks Feasible in Shared, Modern Computing Hardware, International Journal of Organizational and Collective Intelligence ; Volume 8, Issue 2, 2018.

[22] Singh, Shikha, et al. Cloud computing attacks : a discussion with solutions. Open Journal of Mobile Computing and Cloud Computing, 2014, vol. 1, no 1.

[23] Pulkit Sharma. Top 5 deep learning frameworks, their applications, and comparisons !, May 2019.

[24] Amitha Mathew et al, Deep Learning Techniques : An Overview

[25] Gorka Irazoqui et al, Cache Side Channel Attack : Exploitability and Countermeasures, 2016.

[26] Wiki Satat, Neural Networks and Introduction to Deep Learning.

Thèse

[27] Thibaut Probst, Évaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de cloud computing, Systèmes embarqués, INP Toulouse, 2015.

[28] Stephane Fernandes Medeiros, Attaques par canaux auxiliaires : nouvelles attaques, contre-mesures et mises en œuvre.

[29] Gorka Irazoqui, Cross-core Microarchitectural Side Channel Attacks and Countermeasures, WORCESTER POLYTECHNIC INSTITUTE, 2017.

[30] Kahina Lazri,Sécurité de la gestion dynamique des ressources basée sur la prise en compte des profits de consommation en ressources des machines virtuelles, dans un cloud IaaS, Informatique, Université Paris 13.

Mémoire

[31] Bendiab Gueltoum, Sécurité des applications métiers au niveau du Cloud Computing : Contrôle d'accès au niveau des APIs du Cloud Computing, Université Abdelhamid Mehri – Constantine 2, 2015.

[32] Djellalbia Amina, Authentification Anonyme dans un environnement Cloud, Université A.MIRA-BEJAIA, 2015/2016.

Article d'actes de conférence

[33] Abid Shahzad et al, Virtualization Technology : Cross-VM Cache Side Channel Attacks make it Vulnerable, Australasian Conference on Information Systems, 2015.

[34] Ziqi Wang et al, A Shared Memory based Cross-VM Side Channel Attacks in IaaS Cloud, IEEE Conference on Computer Communications Workshops, 2016.

[35] Fangfei Liu et al, Last-Level Cache Side-Channel Attacks are Practical, IEEE Symposium on Security and Privacy, 2015.

Livre

[36] Vimal Kumar et al, Data Security in Cloud Computing, The Institution of Engineering and Technology, 2017, pp 201-213

[37] Seokhie Hong, Side Channel Attacks, Applied Sciences, 2019, pp 99-118

Documents Web

[38] AMAZON WEB SERVICES (AWS) - SERVICES DE CLOUD COMPUTING
https://aws.amazon.com/what-is-cloud-computing/, consulté le 12/02/2021.

[39] HEBERGEURS.TOP
https://hebergeurs.top/cloud-computing-definition, consulté le 12/02/2021.

[40] DIGITAL GUIDE IONOS
https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learning/, consulté le 13/05/2021.

[41] LE DATA SCIENTISTE
https://ledatascientist.com/google-colab-le-guide-ultime/, cosulté le 13/05/2021

[42] MES VMS
https://www.mes-vms.fr/machine-virtuelle-linux-kali-2017-3-light/, consulté le 15/05/2021.

[43] VMWARE
https://www.vmware.com/files/fr/pdf/VMware-Workstation-Datasheet.pdf, consulté le 15/05/2021.

[44] LE LANGUAGE UML ET LE PROCESSUS UNIFIÉ
http://dspace.univ-tlemcen.dz/bitstream/112/5500/5/chapitre1.pdf, consulté le 14/05/2021.

[45] JURI'PREDIS
https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-
consulté le 13/05/2021.

[46] UNIV MLV
http://igm.univ-mlv.fr/~dr/XPOSE2005/XposeVirtualisation/vmware.php#:~:text=L'%C3%A9diteur%20VMWare%20propose%20plusieurs,destin%C3%A9s%20chacun%20%C3%A0%20diff%C3%A9rents%20usages.&text=VMWare%20ESX%20est%20en%20fait,accrue%20de%20la%20machine%20h%C3%B4te., consulté le 15/05/2021.

[47] MACHINE LEARNING MASTERY
https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/, consulté le 13/05/2021.